COMMODORE Disk User

MARCH/APRIL 1989
£2.75

FOR C64 AND C128 USERS

Five drives reviewed

Deciphering machine code

Sector Secrets

Ultima 5 and other Adventures

Speeding up your Sprites

Special Double Sided Issue

CDU Paint Mega Art

The latest software developments & technical achievements

C128 Cannon

THE TOTAL COPY/UTILITIES PACKAGE
CREATED JUST FOR 128 USERS

* Nibbler: Works with single/dual 1541/71 drives
* Fast Copier: Fast data disk backup
* File Copier: Transfers between 1541/71/81
* 1581 Fast Copier: Copy with 1 or 2 1581 drives
* Unstoppable parameter copier
* MFM Copier: Copy unprotected IBM software
* Track & Sector Editor: For 1541/71/81 drives
* Error Scanner: Full featured disk error scanner
* Density Scanner: Checks track densities
* Directory Editor: Reorganize your directories
* PLUS 100 FREE Hot parameters

Your 128 deserves the best
£29.95

New parameter pack contains the latest protection busting code

* Over 400 parameters for the latest software
* For use with Cannon 128
* Deprotect your popular software
* Including EasyScript, SuperBase & Script
* Ideal fc: transffering software to 1581 drive
* Four disk pack with full instructions

From Kracker Jax
£24.95

Oceanic Disk Drive

Now the standard alternative to the Commodore 1541 disk drive

* The Oceanic 5 1/4" disk drive
* 1541 compatible disk drive for the C64
* Loads most software including turbo loaders
* Very quite with external power supply
* External device number select switch
* Direct drive motor, low out-of-alignment risk
* Jiffy DOS installed for fast operating

Ideal for CBM 64 & 128 users
£124.95

Oceanic Plus Renegade
Special Price only £139.95

Slimline 64 Case

Make your old 64 look like a brand new 64C with the Slimline case

* High quality plastic case
* Simple to fit, only a screwdriver required
* Supplied with full fitting instructions
* Low keyboard for easy typing
* New style colours, matches new peripherals

Add style to your old 64
£19.95

Jiffy DOS

Faster than a Cartridge – More powerful than a turbo ROM

Disk Drive Speed–Enhancement System for the 64 and 128

It's faster than the best of the rest & that includes Dolphin DOS

Features Include –

* Ideal for Excelartor, Oceanic, Bluechip, Master 41, 1541, 1571 and 1581 users
* Leaves cartridge and user ports free – Unlike all parellel operating systems
* DOES NOT require additional cables or any hardware between computer and drives
* Guaranteed compatibility with all Software & Hardware – Available for all disk drives
* Chips may be switched off when not required – Should you ever need to
* Simply unplug two chips and plug two in, install two switches – no circuit boards
* 128 Users enjoy your extra power – Jiffy DOS works in both true modes – Allows multiple drives
* 14 Extra DOS commands – including many utilities – easily accessed
* Uses stanadard formats – All existing disks are 100% compatible
* State computer and drives when ordering

The Speeds are impressive –

* Load 202 Blocks in 8 seconds – C1541 loads in 128 seconds – Parellel systems 15 seconds
* Save 202 Blocks in 30 seconds – C1541 saves in 150 seconds – Parellel systems 98 seconds
* Format 35 Tracks in 21 seconds – C1541 formats in 90 seconds – Parellel system 24 seconds
* Load 125 Block Sequential file in 15 seconds – C1541 loads in 84 seconds
* Save 100 Block Sequential file in 24 seconds – C1541 saves in 81 seconds

The most exciting feature is the price
64 and drive £54.95 128 or 128D & drive £54.95
Each extra drive £24.95

Renegade

The most powerful disk backup package on the market today

Utilities at the edge of possibility – at your fingertips

Some of the many features –

* Single or Dual high speed data copier – Single drive 3 pass 120 secs – Dual drives 1 pass 40 secs
* Single or Dual State of the Art Copier – Auto detects protection used & selects required copier
* Single or Dual Fast file copier – Using direct access commands this unique style copier is very fast
* GEOS Modual – Copy GEOS 64/128 (1.3 & 2) all most applications plus a work bench File Copier
* Capable of copying RapidLok protection
* Directory Editor – Organize your disk directories quickly and easily
* Error scanner with unique sector editor PLUS High speed Byte Pattern Scanner
* Scrolling Machine language monitor with disk drive monitor
* Over 300 parameters for those tough programs
* Don't mess with cartridges settle for the best

Customer Support –

* Parameter updates released every 2 months price only £9.95
* Regular Renegade updates four time a year – We will be ahead of the rest – Only £12.95
* Full technical telephone support – of course

Please note we strongly condone all software piracy however legitimate users have a right to backup their software
£24.95

F.S.S.L
FINANCIAL · SYSTEMS · SOFTWARE · LIMITED

● Order with Access, VISA, cheque or Postal Order, COD and Express delivery available, foreign orders add £2.50 per package. Prices include VAT and postage. Please allow 14 days for delivery.

Access VISA

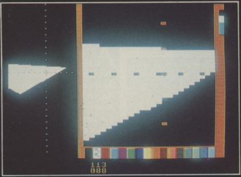
18 High Street · Pershore · Worcs · WR10 1BG · Tel: (0386) 553153

Commodore Disk User
Volume 2 Number 3
March/April 1989

IN THE MAGAZINE

ON THE DISK

CDU Paint



Editor: STUART COOKE
Deputy Editor: FIN FAHEY
News Editor: ERIC DOYLE
Software Evaluation: PAUL EVES
Artwork & Blake's 7 consultancy: ALAN BATCHELOR
Photography: MARK WARFORD
Adventure Correspondent: GORDON HAMLETT
Designer: KIM GOODHEW
Death Threats: PAUL WHITINGTON
Advertisement Manager: PAUL KAVANAGH
Copy Control: ANDREW SELWOOD
Origination: EBONY TYPESETTING
Distribution: S.M. DISTRIBUTION
Printed by: CHASE WEB, PLYMOUTH

Special note: This issue's disk is double-sided to give almost twice as much value for money than usual. Please check the disk instructions page for further information.

Commodore Disk User is a bi-monthly magazine published on the 3rd Friday of every alternate month. Argus Specialist Publications Limited, Commodore Disk User, 1 Golden Square, London W1R 3AB. Telephone: 01-437 0626 Telex: 8811896

Opinions expressed in reviews are the opinions of the reviewers and not necessarily those of the magazine. While every effort is made to thoroughly check programs published we cannot be held responsible for any errors that do occur.

Update

Gender Benders

The Organisation Against Sexism in Software (OASIS) has been formed by Sandra Vogel to combat sexual bias and exploitation wherever it raises its ugly head in computing. In the eyes of OASIS, Palace boobed with Maria Whittaker, US Gold showed an indiscrete amount of male chauvinist piggery with their recent advertising campaign and various companies have exposed their position with strip poker games galore.

Any move against overt sexism is to be welcomed but where should the line be drawn? Should software houses adopt the extremely patronising stance of Audiogenic who promise positive discrimination towards women? On the other hand, can we rely on software houses to exercise discretion?

Watchdog organisations are fine but rarely achieve success against market pressures. The software industry is a male oriented market because this is where the interest and the money lies.

OASIS has a long, hard battle to fight if it intends to eliminate heroes. Software reflects the real world in which the *Sunday Sport*, *The Sun*, *Rambo*, Indiana Jones and Barbara Cartland unfortunately prove so popular. Good luck, OASIS.

An annual subscription to OASIS costs £3 and includes six issues of the bimonthly OASIS magazine.

All enquiries and payments should be directed to Sandra Vogel, 3 Alden Court, Stanley Road, Wimbledon, London SW19 8RD.

Alan Clark cashes in on Summit's competition

Free LSD

As Dennis Norden might say, "Do you remember the days when an LSD experience was called pay day?" Alternative Software's boss, Roger Hulley, certainly does because his new label, Summit, is named after a game, devised by his father, based around the old British currency.

To mark the launch of the £2.99 budget label last summer a simple competition was devised in which the entrants had to calculate the cost of a Summit game in old money. The prize was a full set of mint coins from the old days and Alan Clark of Camberwell was the sender of the winning entry.

Future plans for Summit include the release of *The Double*, Goliath Games' (previously Johnson Scanatron) blockbusting football game, followed by Database's highly successful *Mini Office* package containing wordprocessor, database and spreadsheet.

Electronic Aids

Electronic Arts' customer services have a pile of clue books for perplexed players of the *Bard's Tale* series, *Deathlord*, *Wasteland*, and *Mars Saga*.

All of the books contain essential hints and tips alongside detailed maps and mazes designed to help the serious roleplayer to reach their goal.

Each clue book costs £5 from Customer Services, Electronic Arts, Langley Business Centre, 11-49 Station Road, Langley, Berkshire SL3 8YN. Tel: (0753) 46465.

Compumart Pack It In

Compumart are offering a starter pack worth £30 with Amstrad, Citizen and Panasonic printers.

The starter packs contain 1000 sheets of listing paper, an extra ribbon and an interface cable for the selected printer. The printer range, priced from £139 plus VAT, includes 9 and 24 pin models with 80 and 132 column versions. Credit card owners can ring the Compumart Hotline on (0509) 610444 and take free delivery of a printer and starter pack on the next working day. Other customers will get a similarly speedy reply from Compumart Ltd, Jubilee Drive, Loughborough, Leics LE11 OXS.

Gulf Wars

Although diplomatic oil has been poured on the troubled waters of the Arabian Gulf, Again Again has used the American/Iranian strife as the basis for their latest release, *Operation Hormuz*.

Programmed by Durrell Software, the game involves the player as an American VTOL pilot whose mission is to destroy heavily defended oilfields while ensuring the safety of his US aircraft carrier base.

***Operation Hormuz* is available for the Commodore 64 and costs £9.99.**

Epson Moves

Epson UK, who claim 40 per cent of the British dot matrix printer market, have moved from Wembley to new premises in sunny Hemel Hempstead.

The new complex offers Epson the opportunity to provide better facilities for training supported by a projected audio visual theatre and demonstration areas.

The full address for Epson (UK) Ltd is Campus 100, Maylands Avenue, Hemel Hempstead, Herts HP2 7EZ. Tel: (04222) 61144.

Active Vision

After the Mediagenic fiasco, renamed Activision is now busily signing new software houses to its label.

The name of David Crosweller was respected in the industry long before he joined Infogrames but since his departure from the company he has been keeping a low profile while busily negotiating a deal with Activision's European Vice-President, Hot Rod Cousens.

Crosweller's development company, New Frontier Productions, has just been signed a two year contract to product a range of titles for Activision.

The fruits of this collaboration should start to appear in late summer.

Two more companies have also been affiliated with Activision: Vivid Images and Motion Picture House.

Vivid Images is the collective name chosen by John Twiddy, Hugh Riley and Mev Dinc who have all previously worked on System 3 projects such as *Last Ninja II*. As independent programmers, their identities were often submerged behind the commissioning companies names but it is hoped that the new corporate title will bring them the recognition they deserve.

Although Motion Picture House will be producing original concepts, plans are afoot to participate in developing products for the Nintendo Entertainment System. Initial releases for this year will feature the work of Jonathan Griffiths, Glynn Williams and Benni Notoriani.

David Crosweller explores New Frontier for Activision

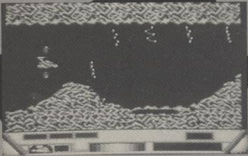
Fantastic Journey

Cascade Games latest product, *DNA Warrior*, is reminiscent of the movie *Fantastic Voyage* in which a submarine crew were miniaturised and injected into someone's bloodstream.

In *DNA Warrior*, the player has to destroy a second brain which is growing inside a scientist whose implant experiment is going very wrong. Piloting a microsubmersible through the bloodstream is a dangerous battle against the body's immune system and mutant brain tissue as the craft passes through bionic limbs and a pacemaker on the way to the brain.

The disk version of Cascade's game is available from mid February and costs £14.95.

Explore the inner-self with Cascade's DNA Warrior



Welcome

Once again, this is a bumper double-sided issue of *Commodore Disk User*. Unlike our Christmas issue, this time we concentrates on utilities, rather than games. On the flip side of the disk we have an all-singing, all-dancing graphics package from well-known programmer Tony Crowther. Tony's crammed every feature he reasonably could into *CDU Paint*, and the result is a program of commercial standard. On our A side we are featuring one of the biggest Basic extension utilities in the known universe. Paul Eves, one of our regular contributors, has supplied *Devaid*, no less than 41 extra Basic commands – a boon to those sick of the C64's limited Basic. C128 owners are catered to by an 80-column graphics package, and we've added a full range of games.

Besides our feature programs, we are now starting an interactive series by CDU regular Eric Doyle which will teach you machine code painlessly over the coming issues, and Allen Webb continues with his suite of graphics routines.

One small apology is needed. Last issue's C128 *Menu Maker* program is missing a small routine which allows you to merge on Side A of the current disk. To use it see the instructions in the last issue.

Fin Fahey

How to copy CDU files

You are welcome to make as many of your own copies of Commodore Disk User programs as you want, as long as you do not pass them on to other people, or worse, even sell them for a profit.

For people who want to make legitimate copies, we have provided a simple machine-code file copier. To use it, simply select the item FILE COPIER from the main menu. The copier works with a single drive, is controlled by means of the function keys as follows:

F1: Copy file – the program will prompt you for a filename

F3: Resave the memory buffer – you may get an error on a save (perhaps you left the drive door open). Use this to try again.

F5: Disk commands – allows you to enter any regular C64 disk command

F7: Displays the directory

F2: Exits the program and returns you to Basic.

Disk instructions

We have done our best to make sure that Commodore Disk User will be compatible with all versions of the C64 and C128 computers

Getting the programs up and running should not present you with any difficulties, simply put your disk in the drive and enter the command:

LOAD "MENU",8,1

Once the disk menu has loaded you will be able to start any of the programs simply by pressing the letter that is to the left of the program you want.

C128 users please note that you should be in C64 mode when using the disk. You can enter C64 mode by either:

i) Holding down the Commodore key (bottom left of the keyboard) when turning the computer on or,

ii) After turning the computer on type GO64 and answer "Y" when prompted "ARE YOU SURE?".

IT is possible for some programs to alter the computer's memory so that you will not be able to LOAD programs from the menu correctly until you reset the machine. We therefore suggest that you turn your computer off and then on before loading each program.

Disk Failure

If for any reason the disk with your copy of Disk User will not work on your system then please carefully re-read the operating instructions in the magazine.

If you still experience problems then:

1) If you are a subscriber, return it to:
INFONET LTD
5 River Park Estate
Berkhampstead
Herts. HP4 1HL

2) If you bought it from a newsagents, return it to:
CDU Replacements
Direct Disk Supplies
Unit 19
Teddington Business Park
Station Road
Teddington
Middx TW11 9BQ
Telephone: 01 977-8777

Within eight weeks of publication date disks are replaced free.

After eight weeks a replacement disk can be supplied from DDS for a service charge of £1.00. Return the faulty disk with a cheque or Postal Order made out to DDS for £1.00 and clearly state the issue of CDU that you require. No documentation will be provided.

Please use appropriate packaging, cardboard stiffener at least, when returning a disk. Do not send back your magazine – only the disk please.

Back Issues

Back Issues of Commodore Disk User are available at £3.00 per issue, via:

Infonet Ltd.
5 River Park Estate
Berkhampsted
Herts HP4 1HL

Those magazines available are:

Jan/Feb 1988: Utilities Disk Librarian, Disk Mate, Text Cracker, Noluxe Paint, C128 Ram disk. Games – Five-up, Quad plus Micronet demos.

May/June 1988: Utilities – DrumSynth, Basic tokeniser, C-CAD, Basic compactor, C128 Windows. Games – Santolus, Atlantis

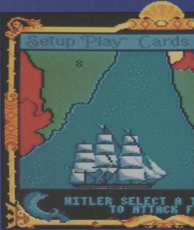
July/August 1988: Utilities Disk Toolkit, Relocator, Orrery, Message Construction Kit. Games – Mind Games, 3D Breakout, Peggy 128

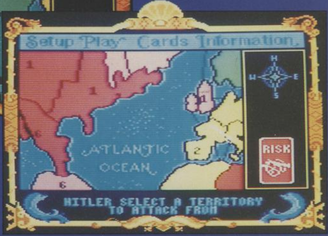
September/October 1988: Utilities Fractal Frolics, Location Finder, Score Keeper, Colour Match, C128 Spreadsheet. Games – Scorpion, Escape, Starburst, Addit

November/December 1988: Utilities CDU FORTH, Texted, Extractor, Windows 64, ZMON 128. Games Oblivion, Cribbage Master.

January/February 1989: Utilities Easy Scroller, Data Maker, Border Sprite, Disk Turbo, Menu Maker 128. Games – Blastball, Microdot, Runaway, Colour Bind, Logic, Spots, Life.

Reviews





RISK

When I rule the world...' How many of us have never suffered from delusions of megalomania? *Risk* is a version of the classic board game that at least gives you the chance to compete for world domination. Prove yourself to be a latter day Napoleon or Attila the Hun and at least you will be able to gloat insufferably over your vanquished rivals.

In case you have never played the original game, *Risk* is definitely not a game to be played with Granny after Christmas dinner. You need to be totally ruthless if you want to win. And people who only play games for fun and not to win definitely shouldn't attempt this one.

Two to six people can play the game although you need at least three players for a decent game. Immediately, the first advantage of the computer version becomes apparent. You don't need any one else to play; you can use computer opponents instead, setting each of them up on one of three different skill levels.

The board is a much simplified map of the world, divided up into some forty odd territories. Territories are distributed at the start of the game either randomly or by choice depending on which set of rules you are playing to. One of the problems with the original board game was that the rules varied considerably on either side of the Atlantic. Again, no problem here. You can play either of the two main versions or customise the game how you see fit.

At the start of each turn, a player receives a number of extra armies depending on the number of territories held. There are also bonus armies for occupying a whole continent and for cashing in sets of cards (you receive one card per turn providing that you have captured at least one territory). These extra forces can be deployed where-ever you see fit.

Combat is simply resolved by throwing dice. The attacking country can roll up to three dice providing he has sufficient armies on a territory, the defending country up to two. the dice are compared and the losing country has to remove one of his armies. This continues until the attacker decides to stop or the defender has no armies left in which case he loses possession of the territory completely and it is occupied by the conquering forces.

A winning strategy needs a careful combination of aggression and defence. Ideally, you want all your forces massed round your borders but in practice, this is seldom possible. That is why Asia is such a difficult continent to defend. Too many countries border onto it.

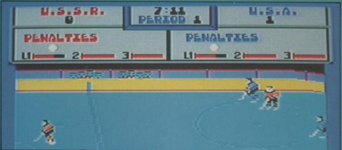
The screen is bright and colourful with good clear graphics and an easy control method. Even if some of the scrolling is a bit slow and jerky, this doesn't detract in the slightest from the game's excellent playability. This is far and away the best conversion of a board game yet seen. **GRH**

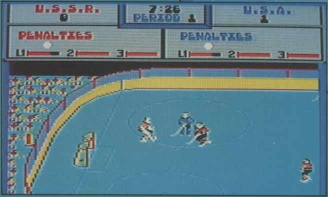
At a glance

Title: Risk
Supplier: Virgin Games, 2-4 Vernon Yard, Portobello Rd., London W11 2DX
Tel: 01-727 8070
Graphics: All the best bits should be coloured pink – Victoria R
Playability: Invading Russia was never this easy – N. Bonaparte
Addictiveness: Vorsprung durch technik – A. Hitler

Power Play Hockey

With only seconds to go the USA scored the winning goal and defeated the Soviet Union to take the 1980 Lake Placid Olympic gold medal. To many this feat will have faded in the mists of time but to the inhabitants of the US of A it lives on and on. So much in fact that nearly nine years later Electronic Arts decide to base its ice hockey game on, and only on, that single match.





To the rest of us who live on this side of the pond still looking for a good ice hockey game that captures the excitement, skill and let's be honest, violence of the world's fastest team sport will see this as a weakness instead of a strength. Having said that, this is probably best ice hockey game you can buy, so far.

The game can be played against human or computer opposition and as in all sports games two player action tends to be better. You can also determine the level of player from junior to Olympic, the time in each of the games three periods that can range from a brisk two minutes to a gruelling twenty and whether it will be a one to one (plus goalminders) or full six-a-side game.

Whatever you decide the match is played on a scrolling 3D ice rink that's complete in every detail including the slightly faded appearance of the blue centre line. The teams skate up to the centre line and the referee drops the puck to begin the match.

You play the centre – the guy that's supposed to score the goals – so you'd better find your form quickly or you'll let down the free world or red menace depending on which country you play. Here's no sign of *glasnost* in this game!

Through standard joystick controls you can skate around the ice but remember that's it's almost impossible to do a 360 degree turn when you're skating on ice at full speed. You can also learn and perfect the moves to try a wrist or slapshot or even drop pass the puck back to a team-mate following you. While on defence you can try to poke your stick to ease the puck away, slide in for a tackle or if that doesn't work try a body check.

Naturally, the referee objects to such behaviour as will your opponents and so if you're too heavy the gloves will come off, a fight will start and you'll spend from one to two minutes inthe sin bin.

In a full six-a-side match you are supported in attack by two wingers that will either try and score themselves or pass to you. In defence, left and right defenders will try and foil your opponents attempts to get round the goalminder and to pass the puck up to you or the wingers but obviously you might drop back and help out in defence as well.

You also double up as the manager and decide overall strategy by switching between your three team squads. Each period begins with a balanced team on the ice but these will quickly tire so it's up to you to decide whether to leave them on or swap them (by pressing a key) for either a squad that's good at scoring goals but poor in defence or an all out defensive unit. This can be vital as in a 20 minute period you could tire out two squads and be left with the defensive unit.

Add to that full game statistic showing the total shots and goals for each player during the game and graphics screens celebrating your victory in either American or Russian and you have the formula for a great game. Unfortunately, you can only replay the 1980 ice hockey final a limited number of times before it goes cold. With a little more variety and perhaps either an international tournament or a league it could have been perfect.

TH

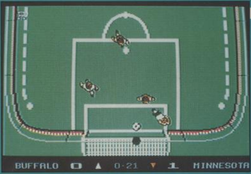
At a glance

Title: Power Play Hockey
Supplier: Electronic Arts, Langley Business Centre, 11-49, Station Road, Langley, Nr Slough, Berks., Sl3 8YN.
Price: £14.95
Graphics: You can almost feel the chill of the ice and heat of the action.
Sound: Cheers and whistles.
Playability: Hard hitting action.
Addictiveness: For a limited time.

Microprose Soccer

There have never been so many football games with Gary Lineker starring in three, Peter Beardsley, Emlyn Hughes and even Roy of the Rovers in others as well as *Football manager II* and a host of budget cassette games. More and more software houses join in with the chant here we go, here we go, here we go...

Even Microprose, which is better known for its high flying simulations, is joining in the action and has signed up a pair of twin strikers from development house Sensible Software as it presents two games for the price of two.



By today's standards, £19.99 is expensive even for a disk game and so *Microprose Soccer* will have to be good to survive. So this game of two halves features on one side of the disk a full 11-a-side game and on the other side an American style indoor six-a-side version.

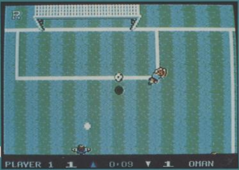
Microprose has opted for a top down view of the pitch which isn't an angle the great football watching public is used to but it works quite well and with a bit of practice you'll find it easy to play with. As in most action style football games you play the footballer with or nearest to the ball and through a selection of joystick moves can dribble, pass and shoot. Like the view of the action some of thse moves will take some practice to perfect. For example, to try a banana shot which is the only realistic chance of beating a computerised goalkeeper you must press the fire button while holding the joystick diagonally backwards either left or right for a left or right shot. This isn't the sort of move that can be guaranteed. Apparently, you can also try a volley, a chip shot or the truly impossible, Pele style, overhead kick.

Whenever the ball enters the box a buzzer sounds telling you that you know control the goalkeeper who must simply touch the ball to catch and control it. Unfortunately, it would be better if you chose when to swap control as this automatic change can cause problems. For example, when you're defending you move players across to tackle the opposing forward but in mid manoeuvre the buzzer sounds swaps control to the keeper and suddenly the joystick moves that would have moved a defender into position to tackle moves the keeper out of the goal leaving the net wide open and you with no chance of recovering.

Tackling consists solely of the sliding variety which can be dramatic if they work and disastrous and embarrassing if they miss. This is exaggerated on a wet pitch when it rains heavily (a new feature that will copied by others) as a missed tackle can cause a defender to literally get in a spin and zoom off the pitch.

You can play *Microprose Soccer* games either as one off matches or as part of a world cup tournament with six groups and seeded teams or you can take part in the Microprose challenge which is a good way to learn the game as you gradually work your way through the 29 teams stored in the game. Starting with Oman you gradually work your way up through Scotland and Denmark until you can take on the likes of Brazil and West Germany on equal terms when you're ready for the World Cup tournament.

If it all becomes too much for you then simply flip the disk and play in a six group American style indoor game with teams like Houston and New York replacing Italy and France. With the smaller pitch and boards to rebound off behind each goal the indoor game is actually a lot faster a may provide a new challenged for even experienced players.



Overall, *Microprose Soccer* is a very good football game but has a few irritating features and an expensive price tag that may deter some, but for dedicated football fans its another one to add to the rapidly growing collection. **TH**

At a glance

Title: Microprose Soccer.
Supplier: Microprose, 2, Market Place, Tetbury, Gloucs, GL8 8DA.
Price: £19.95.
Graphics: Gauntlet meets football.
Sound: more whistles and cheers.
Playability: Some difficult moves to learn.
Addictiveness: There's enough here to drive you back for more.

COMMODORE Disk User

MARCH/APRIL 1989
£2.75

FOR C64 AND C128 USERS

Five drives reviewed

Deciphering machine code

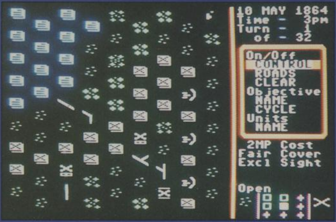
Sector Secrets

Ultima 5 and other Adventures

Speeding up your Sprites

Special Double Sided Issue

CDU Paint Mega Art



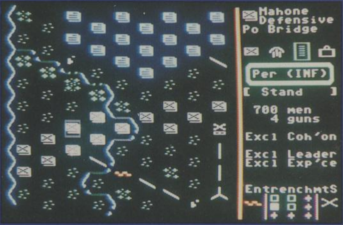
AMERICAN CIVIL WAR VOL III

This is the third and final volume in the *American Civil War* series from the Australian company, Strategic Studies Group. Six battles from the closing stages of the war are presented.

Ironically, three of these battles were tactical victories for the Confederacy but the North had control of all the main supply routes and a had come to terms better with the emerging technology such as the railroads. Nothing at this stage of the war could prevent their inexorable drive south.

The *Wilderness* is the first of the six battles, a confused struggle between Lee and Grant. Grant then tried to outflank Lee and take Spotsylvania. Lee had foreseen this however and had managed to entrench his forces sufficiently that Grant had no option but to lead a bloody frontal assault. A similar situation occurred at Cold Harbour. Grant was convinced that Lee's army was on the brink of defeat and planned his attack carelessly with heavy casualties ensuing.

Atlanta saw the newly appointed General Hood attempting to stop Sherman's march through Georgia. After this attempt had gone with the wind, Hood moved into Tennessee and tried to entrap Schofield's army. The trap was bungled and the two armies met at Franklin where Hood, his judgement clouded, took a heavy beating. Hood was ready to take another heavy defeat at Nashville but thick fog postponed the inevitable. When it cleared the next day, the south was overwhelmed again and the war effectively over.



If, after all that, you still feel in need of further conflicts, then utilities included in the package allow you to design or modify scenarios as you see fit. Suggested variants are included for each battle and there are step by step instructions in the documentation as to how to go about modifying things.

Indeed, the whole of the documentation is first class with detailed historical notes, coloured maps and even spare sticky labels for your scenario disks!

Despite the excellence of this package, this is not really a game suitable for beginners. Experienced wargamers, though, with at least a passing interest in the Civil War will get hours of challenging entertainment. This is how computer wargames should be presented.

GRH

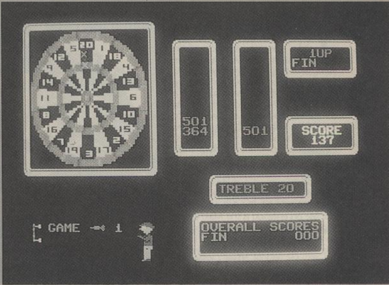
At a glance

Title: Decisive Battles of the American Civil War Volume III
Supplier: Electronic Arts, Langley Business Centre, 11-49 Station Road, Landley Nr Slough, Berks SL3 7YN Tel: (0753 499442).
Price: £18.95
Graphics: Simple but adequate
Playability: Cursor key movement hard to master
Addictiveness: See you at Appomattox

DARTS

Get those arrows ready - that treble 20's waiting!

By Michael Ware



You don't have to make that long cold trip down to the local any more - CDU has added one more facet to the computerisation of society - computer darts.

CDU darts can be played solo - see how quickly you can score 501 - or against a friend. The solo game requires a joystick in port 1, whereas the two-person variant can be played either with the one joystick or two.

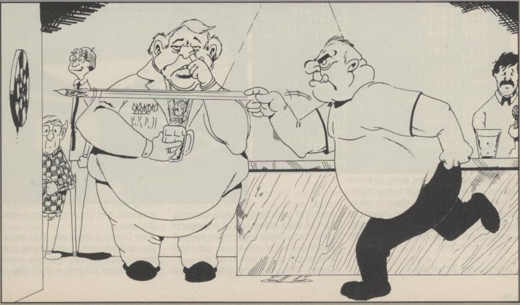
In case there's anyone out there who doesn't know the rules, the game represents the commonest darts variant - 501. The object of this game is to get to a score of 501 as fast as possible. Usually this requires each player to score a double first (dart in the outer ring), but the CDU version doesn't ask for this.

The catch with 501 is, however, that you need a double to close the game. If you only need 10 points, scoring, say, 20 will invalidate your turn. You can, on the other hand, do it with one dart - double five - or by a combination such as a four and a double three.

That's all there is to the rules. In the computer game however, aiming with the joystick could be too easy. To make it hard, the cursor representing your dart wanders all over the place, just like your hand might when aiming. Compensate for this, and hit the fire button when the cross is over the right point, and you might get the score you want. Good shooting!

Loading the program

To load the game outside the menu just enter LOAD "DARTS", 8 and RUN.

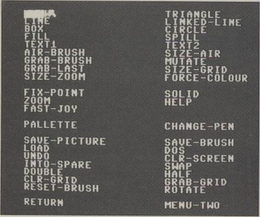


CDU Paint

Take your 64 into a new world of graphics. Tony Crowther's CDU Paint is the only graphics package you'll ever need

You may have heard about some of the incredible art packages that are availabe for 16-bit machines, such as the Amiga's *Deluxe Paint*, and longed to try them out. At last, we can give C64 owners a chance to experience the thrill of using such a package. *CDU Paint* is one of the most advanced art packages ever written for the C64 – but then it *was* designed by Tony Crowther, one of the most experienced Commodore programmers in the business.

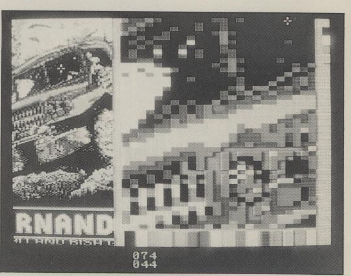
CDU Paint operates in multi-colour mode. It is entered outside the usual CDU menu by simply entering LOAD "CDU PAINT",8,1. The first screen you will see on entering is the Main Menu (Figure 1). All the main functions can be accessed by joystick (in Port 2) from this menu, or via keyboard presses while using the drawing screen. For the moment we will refer to menu functions. A full table of key equivalents for all these commands is given later on.



```
                    TRIANGLE
LINE                LINKED-LINE
BOX                 CIRCLE
FILL                SPILL
TEXT1               TEXT2
AIR-BRUSH           SIZE-AIR
GRAB-BRUSH          MUTATE
GRAB-LAST           SIZE-GRID
SIZE-ZOOM           FORCE-COLOUR

FIX-POINT           SOLID
ZOOM                HELP
FAST-JOY

PALLETTE            CHANGE-PEN

SAVE-PICTURE        SAVE-BRUSH
LOAD                DOS
UNDO                CLR-SCREEN
INTO-SPARE          SWAP
DOUBLE              HALF
CLR-GRID            GRAB-GRID
RESET-BRUSH         ROTATE

RETURN              MENU-TWO
```

The most important thing to know is how to flip from the drawing screen to the menu. This is achieved by either selecting Return from the menu, or hitting the left-arrow key, which toggles back and forth between the menu and screen.

Pens and Brushes

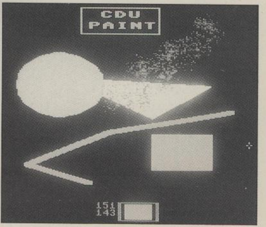


Like its 16-bit cousins, all drawing in *CDU Paint*, whether this be simple freehand sketching or box drawing, can be done using one of a variety of pens, or by designing a brush yourself. Bearing this in mind, we can now discuss the commands in detail.

Draw: This is the most obvious of the commands. Using the current pen or brush, this is a simple freehand draw.

Line: Used for drawing one straight line. One button press defines one line endpoint, then you click again to define the next.

Box: Draws a rectangle. This requires you to firstly click to define one point, then click for the opposite corner. Having sized the box, a third click will



draw it.

Triangle: Similarly to Box, this allows you to set up a base line, and then choose the position of the apex of the triangle.

Linked Line: This is a way of drawing lines that used to be known as rubber-banding. It is used very similarly to the line draw, except that the endpoint of the last line is used as the starting point of the next.

Circle: This is also used to draw ellipses of any aspect. First, the major axis of the ellipse is determined with one click, then the minor axis can be sized with the second click. If you just want a circle, two clicks on the same radius will do the job.

Text1 and **Text2**: Used to put text from the current character sets on to the screen. The current pen is used, and spacer is recognised. Return can also be used to move to the next 'line'. Those are the fundamental drawing commands, but a large array of modifiers and special effects turn them into very powerful facilities indeed.

Special Effects

Solid: This simply ensures that all shapes, such as boxes and triangles are filled with the current colours when drawn.

Fill and Spill: These are two ways of filling an area with the current foreground colours. Depending on the shape of the area to be coloured, one may be faster than the other. Always, however, make sure that the area Filled or Spilled is completey enclosed, otherwise the colour may leak and make a terrible mess.

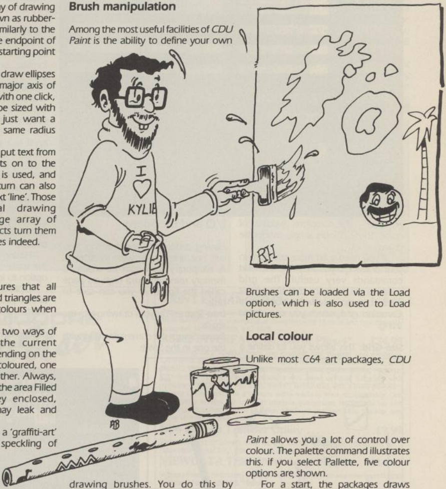
Air-Brush: This produces a 'graffiti-art' effect by spreading a speckling of strokes of the current pen over a wide area. You can decide how big that area is by using:

Size-Air: This command sizes the air-brush area. Use the joystick to vary the radius of the area you need then click to fix it.

Change-Pen: This allows you to flip through a variety of pen forms (see Diagram 2), varying in size and/or number of dots.

Fast-Joy: This toggles between fast and slow joystick movement.

Zoom: With this, you can blow up the drawing area, and get in really close to alter individual pixels. The screen will show a zoom window, plus the part of the screen you are working on at its actual size. A very useful way of tidying up your sketches.



Brush manipulation

Among the most useful facilities of *CDU Paint* is the ability to define your own drawing brushes. You do this by selecting an area of the screen by using:

Grab-Brush: This requires you to define a box. If you click on this, the foreground pattern underneath becomes the current brush. This brush can then be drawn with, modified, or stored. Brush modification commands include:

Mutate: Allows you to resize the current brush box. The brush pattern will be modified to fit.

Double: Doubles brush size.

Half: Halves brush size.

Rotate: Rotates brush by ninety degrees.

Grab-Last: Flips back to the previous brush.

Reset-Brush: Clears the current brush.

Save-Brush: Saves the current brush to disk. Requires you to enter a filename.

Brushes can be loaded via the Load option, which is also used to Load pictures.

Local colour

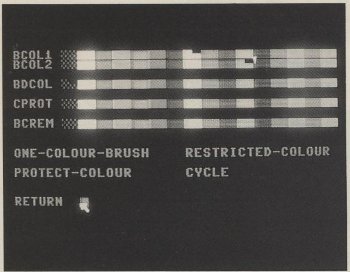
Unlike most C64 art packages, *CDU Paint* allows you a lot of control over colour. The palette command illustrates this. if you select Pallette, five colour options are, shown.

For a start, the packages draws with not one but two foreground colours, since multi-colour mode allows this. You can therefore get a textured colour mix effect. Selecting Brush Colour 1 by clicking on it means that Brush Colour 2 automatically follows, but this can be reselected.

'bdcol' on the palette is not of such importance, but is used as a border colour when saving and loading.

'cprot' allows you a very powerful facility. This defines certain foreground colours as protected *2* you cannot overwrite them.

'bcrem' on the other hand defines certain colours as automatically

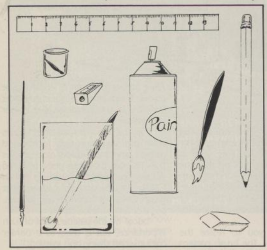


```
BCOL1
BCOL2

BDCOL

CPROT

BCREM

ONE-COLOUR-BRUSH        RESTRICTED-COLOUR

PROTECT-COLOUR          CYCLE

RETURN
```

erasable – the complete opposite of cprot. These colours simply cannot be drawn at all.

If you need a bit more structure in your drawings, you may find the Grid commands very useful. The grid confines the positions that the pen or brush can go to, to positions on a Cartesian grid which you can define using:

Size-Grid: This allows you to define a box which will be the fundamental unit of grid spacing.
Clr-Grid: Removes the grid constraint and allows total freehand drawing.
Grab-Grid: Sets the grid spacing to the brush size.



I/O and the Spare

Having produced your drawing, you will not unnaturally wish to store it. A temporary store can be made to memory using an extra high-res page called the Spare, or you can save to disk.
Into-Spare: Puts your drawing into the spare.
Swap: Swaps the current drawing with the one in the Spare.
Save-Picture: Allows you to save the picture to disk.
Dos: Lets you enter any Dos command – this means that you can erase files, initialise disks and so on. Finally to load a picture in, select the Load menu option 1.

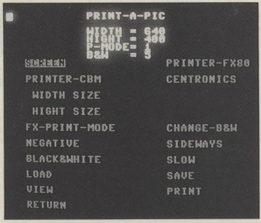
Menu 2

That's not all that *CDU Paint* can do. At the bottom of the Main Menu, you'll see the option Menu-2. This second menu loads in extra programs from disk to allow you to either send your screen to the printer in one of a variety of formats, or to extract a sprite pattern from the screen.
Option 1, Hires-edit, is the main drawing screen.
Option 2, Hires-Sprite, gives you the sprite extractor screen. This shows you the current picture, but superimposed on it is a box which you can use to extract a sprite at the press of a button. Pressing L will toggle between being able to move this box fast or slow.

You can select which sprite position the sprite will enter by using the + and – keys, and the sprite colours can be varied by using the numbers 1-4 for sprite colours 1 and 2, background and border.

Hitting Run/Stop will exit you from the sprite utility and will give you the option to load or save sprites.

Option 3 on menu-2, Hires-Priter, allows you to get Hard Copy of your masterpiece. Unlike the main drawing screen, you can load pictures from other drawing utilities using this menu, including Blazing Paddles, Koalapaint and Paint Magic.



```
          PRINT-A-PIC
          WIDTH  = 640
          HIGHT  = 400
          P-MODE= 1
          B&W    = 5
SCREEN                      PRINTER-FX80
PRINTER-CBM                 CENTRONICS
 WIDTH SIZE
 HIGHT SIZE
FX-PRINT-MODE               CHANGE-B&W
NEGATIVE                    SIDEWAYS
BLACK&WHITE                 SLOW
LOAD                        SAVE
VIEW                        PRINT
RETURN
```

The Print menu allows you to use either a CBM printer, an Epson FX80 or a notional Centronics standard (we cannot guarantee that this will work with your particular printer, however). It also allows you to determine the print-out size in pixels and determine whether or not this will be normal or sideways printed. Other options include Negative printout, and black-and-white positive.

Key Summary		Q:	Rotate brush
		R:	Cycle colour
		S:	Solid on/off
A:	Text1	T:	Flip restricted-colour
Shift/A:	Text2	U:	Undo/swap
B:	Box	V:	Triangle
C:	Circle	W:	Flip one-colour-brush
D:	Draw	X:	Define grid
E:	Exchange with spare	Shift/X:	Put object
F:	Spill	Y:	Choose pen
Shift/F:	Fill	.:	Force colour (change background)
G:	Adopt colour under cursor	-:	Halve object
H:	Help grid	+:	Double object
I:	Air-brush	@:	Pause cycle
Shift/I:	Size air-brush	f1/f3:	Change colours 1 & 2
J:	Flip joystick mode	f5/f7:	Change colour 2
K:	Linked-lines	Run/Stop:	Draw
L:	Line	1-8:	Set grid
M:	Mutate brush	Home:	Save to Spare
N:	Grab brush	Clr:	Clear Screen
Shift/N:	Last brush	Space:	Zoom
O:	Flip protect-colour	Shift/Space:	Define Zoom
P:	Pallette	Left arrow:	Main menu

Be creative

We're sure that you'll get an awful lot of fun from *CDU Paint* , and find it extremely useful too. To show what you can do with it, we've included a number of pictures on the disk, some of which can be shown as a slideshow by entering LOAD "SLIDESHOW",8 and RUN.

Finally a last note on *CDU Paint* file formats. All screens from the utility are runnable files, so you don't need *CDU Paint* to view them. Just load the screen and Run. ■

Commodore Key commands:	
L:	Load
S:	Save picture
N:	Save brush
D:	Dos
S:	Save to spare
P:	Pallette
N:	Reset brush

COMMODORE *Disk User*

Coming soon...

CDU GOES SONIC

In the next issue of Commodore Disk User, we'll be looking at sound on the C64.

IN THE MAGAZINE

Frankfurt Music Fair report
MIDIing up the 64
How to be a computer composer
Programming SID

Plus...
Instant sound software on disk, together with our usual range of games and utilities. The May/June issue of CDU will be on sale in the third week of April. Don't miss out!

Devaid

Soup up your Basic with our powerful Development Aid

Devaid, which is short for Development Aid, adds 41 commands to the C64 Basic. It gives the programmer a very powerful army of new commands to aid the development of programs written in Basic. Some of the commands can be used in program mode as well as direct mode, provided the toolkit is resident in memory.

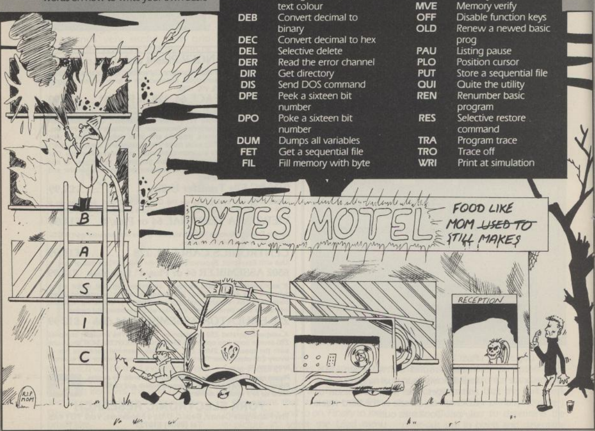
To give the programmer optimum memory available, the toolkit sits between $8000 and $A450. (Decimal 32768 to 42064). To minimize the typing, all the commands use just three letters. For those of you that have used my Disk utility and Disk toolbox, some of the commands will be familiar.

Instead of writing thousands of words on how to write your own Basic commands, (This topic as been covered many times before), I will go straight into the commands, their meanings, and how to use them.

I will now show how to use each of the new commands in turn.

APP: Appends the named prog to the one you already have in memory. APP"program name",8

Command	Meaning
APP	Append files (Basic or M/C)
AUT	Auto line numbering
BIN	Convert binary to decimal
CHA	Load and run a Basic prog
CHG	Change given string
COD	Replace graphic symbols on list
COL	Change Border/Screen/text colour
DEB	Convert decimal to binary
DEC	Convert decimal to hex
DEL	Selective delete
DER	Read the error channel
DIR	Get directory
DIS	Send DOS command
DPE	Peek a sixteen bit number
DPO	Poke a sixteen bit number
DUM	Dumps all variables
FET	Get a sequential file
FIL	Fill memory with byte
FIN	Find given string
HEL	Display help menu
HEX	Convert hex to decimal
HIM	Set top of memory
INP	Selective input
KEY	Display function key settings
LOM	Set bottom of memory
MEM	Display free memory available
MER	Merge programs
MLO	Memory load
MSA	Memory save
MVE	Memory verify
OFF	Disable function keys
OLD	Renew a newed basic prog
PAU	Listing pause
PLO	Position cursor
PUT	Store a sequential file
QUI	Quite the utility
REN	Renumber basic program
RES	Selective restore command
TRA	Program trace
TRO	Trace off
WRI	Print at simulation



AUT: Automatic line numbering. AUT10,10 will start numbering from 10 with an increment of 10. AUT on its own will disable the command. Make sure that the second parameter does not exceed 255.

BIN: This command converts an eight bit binary number to two decimal numbers, the high and low byte values. Any number of conversions can be catered for. Simply separate them with a comma.

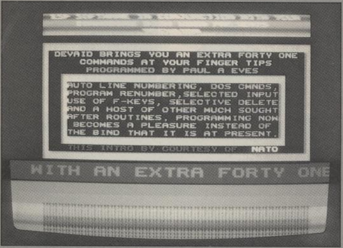
Example: 00011101 or 01001101,11110001

CHA: Loads and runs a Basic program. Resetting all pointers after the load. Used the same as the normal LOAD command.

CHG: Changes all occurances of the given string or command to the new string/command. CHG,INPUT,,PRINT, will change all occurences of INPUT to PRINT.

COD: This handy routine allows you to change all those unreadable graphic characters, into clear readable mnemonics. To use you must have a program in memory. Type COD, then when you list the program, it becomes readable.

COL: Allows easy manipulation of the screen, border and text colour. No more 'Poking' around. Simply type COL2,3,7 (Or similar).



```
DEVAID BRINGS YOU AN EXTRA FORTY ONE
    COMMANDS AT YOUR FINGER TIPS
     PROGRAMMED BY PAUL A EVES

AUTO LINE NUMBERING, DOS CMNDS,
PROGRAM RENUMBER,SELECTED INPUT
USE OF F-KEYS, SELECTIVE DELETE
AND A HOST OF OTHER MUCH SOUGHT
AFTER ROUTINES, PROGRAMMING NOW
 BECOMES A PLEASURE INSTEAD OF
THE BIND THAT IT IS AT PRESENT.

THIS INTRO BY COURTESY OF NATO

WITH AN EXTRA FORTY ONE
```

DEB: Works the same way as the BIN command, except the input number is Decimal. The binary equivalent is returned.

DEC: As DEB and BIN except the Hexadecimal equivalent is returned.

DEL: Delete specified single or multiple lines. DEL 1000,2000 would delete lines 1000 to 2000. DEL3000 would delete all from line 3000 onwards.

DER: This command simply reads the error channel, reporting any problems.

DIR: Reads the disk directory without spoiling the program in memory. This command caters for dual drive users, therefore you need to add either a 0 or a 1 depending upon the drive number you are using.

DIS: No more opening, printing and closing of files with this command. Simply type DIS "command:string1, string2, etc". Example: DIS"RO:fred=bill" will rename the prog bill to the new name of fred.

DPE: Standing for Double-Peek. This will return a sixteen bit decimal number from address given. Example PRINT DPE(43) returns the sixteen bit number held in address 43 and 44. Because DPE is a function, you must remember to preceed it with PRINT.

DPO: Same as DPE except you are Poking a sixteen bit number. (No need to precede it with print obviously).

DUM: Once a program has been run, this will dump out all the variables.

FET: Fetches the named Sequential file and loads it into memory. FET"source code" will load the Seq file source code into memory.(Disk only).

FIL: Fills specified memory with given byte. FIL 4000,8000,69 will fill from 16384 to 32768 with ascii code 69 (E). If filling screen memory, remember to use the screen codes.

FIN: Used in conjunction with CHG. This will search for and report all occurances of the given string/ command. Example: FIN print will report all occurences of the word print.

HEL: Should you forget all the commands that are available, HEL will simply display a list of all of them.

HEX: Same as the other number convertors except the decimal equivalent is shown. Note do not prefix with the dollar sign. Eg: HEX COOO.

HIM: Sets the top of Basic memory. HIM 32768 sets the top of memory to 32768.

INP: Enables you to use a specialised form of INPUT command. Allows you to specify the position of the input without the flashing question mark. EG-INP(20,10)"What is your name"; var.

KEY: Used on its own, KEY will display all the function keys and the text that is stored on them. (All sixteen keys are already programmed for you). You can change the default settings by using KEYnumber,"text". For instance KEY1,"AUT05,5" will assign the auto line numbering command to function key 1. By placing a back-arrow before the closing quotation marks, you will instruct the computer to perform a carriage return.

LOM: Sets the lower limit for Basic memory. Used the same way as HIM.

MEM: Displays the total number of free bytes remaining. (Gives the true number). To use simply type MEM on its own.

MER: This is a more powerful command than APP. This will merge two programs in such a way that all the line numbers will be in sequence. In other words, it doesn't just tack the program onto the end of the other. If the programs have the same line numbers, then the merging program takes priority.

MLO: Allows you to load a block of memory from disk. You specify the address you wish to load to.

MLO"name",dev,1,address. (Note: do not forget to add the 1).

MSA: As MLO except you SAVE out a block of memory. The ending address plus 1 must also be used. MSA"name", dev,1,sa,ea+1.

MVE: As the above except you are Verifying.

OFF: Once you no longer need the preset function key commands, you turn them off by typing OFF.

OLD: Should you accidentally NEW while your Basic program is in memory, then simply type OLD to get it back.

PAU: Trying to read basic lines as they flash passed you after listing is somewhat tiresome. To overcome this, before you LIST type PAU. Now when you list you can pause the listing by pressing the shift key. (Shift/lock will pause until released).

PLO: Short for Plot, this command allows you to position the cursor anywhere on screen. Used as main routine for WRI & INP commands. PLO(20,10) will position the cursor at row 20 column 10.

PUT: Saves out a sequential file to disk. Handy when using source files that are saved as seq-files.

QUI: When you have finished using the utility, QUI will return you to normal. SYS32768 will re-activate.

REN: When developing large basic programs, it will become necessary to add or delete sections of line numbers. This can become very awkard indeed. REN allows you to renumber your program, taking into account all GOTOs and GOSUBs. The syntax is RENx,y,z where x is the line number to start from. y is the increment and z is the new start number. Note if x is 0 then the whole program is renumbered.

TRA: This command lets you trace through a program a line at a time. Once activated you must press the space bar. The line number is displayed at the top of the screen. By hitting a numeric key from 1-9 whilst in trace mode, will speed up or slow down the trace.

TRO: Disables the trace command. (Do not type TRO if you haven't already initialised the trace.

WRI: Allows for customising your print statements. This command simulates the 'print at' command. Syntax is WRI(15,10)"string".

I have not done any fancy coding routines. Therefore, those of you that are interested, can easily follow the code using a monitor. One point I must make is this. Certain cartridges and memory resident fast-loaders, may play havoc with some of the routines. If this happens just remove them and start again. Happy developing.....

Bazair

Can you survive the quest for the hidden keys?

It is said that in the Bazair system, somewhere on the Galactic Rim, there are nine planets, each more inaccessible than the last. On each planet is a city, and in each city is an electronic key of such a nature that it can be used to command the ship that will carry the fortunate traveller to the next city inwards. It is also said that in the ninth city will be found the key to the mastery of all time and space.

'None have come into possession of the secret, for many perils encompass the cities and planets. The key to each city is in each case protected by fast-moving robot guardians. Should a questing hero wish to remain non-posthumous, they were wise to step swiftly.

'Once in possession of the key, the voyager's problems will start. For each of the craft that will be needed to penetrate further into the system is protected by a lake of mineral acid so corrosive that it will reduce a living being to a solution of ions in milliseconds. A network of floating platforms passes over the dread lake, and swiftness and dexterity will be needed to master the pattern of their movement.

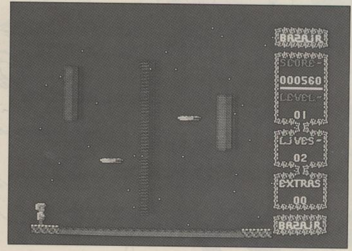
'But stay – pity the unfortunate venturer who travels then inwards on the ethereal winds. For between each of the planets is a zone of asteroids of unusual density and possessing most unpredictable orbits. Few ships that venture into the system are ever heard of again, and we can only conjecture that they are now little more than mangled debris in the belts.

'Are the wretched seeker's troubles now over? Not so – for each planet possesses a gravity markedly higher than the last. Landing by means of the retro jets becomes a hazardous affair before long.

'But finally, the wanderer may at last enter the appropriate city. Alas, such a fate is far from enviable, for the cites will not part easily with their unbidden guests, and are wont to toy with them. It is said though that a determined and wise person may master the mazes and thus obtain the next key.

'It is inconceivable, however, that any should master all the rings, and many have sought to try. Perhaps it is as well for the universe that this is so.'

Extract from *The Book of Half-truths,* Baron Bodissey (Alphanor Press, 3017)



Controls

To get key: Joystick left/right
To cross lake: Joystick left/right, fire to jump
To dodge asteroids: Joystick left/right
To land on planet: Joystick left/right, fire to ignite retros
To negotiate maze: Joystick left right, fire to jump

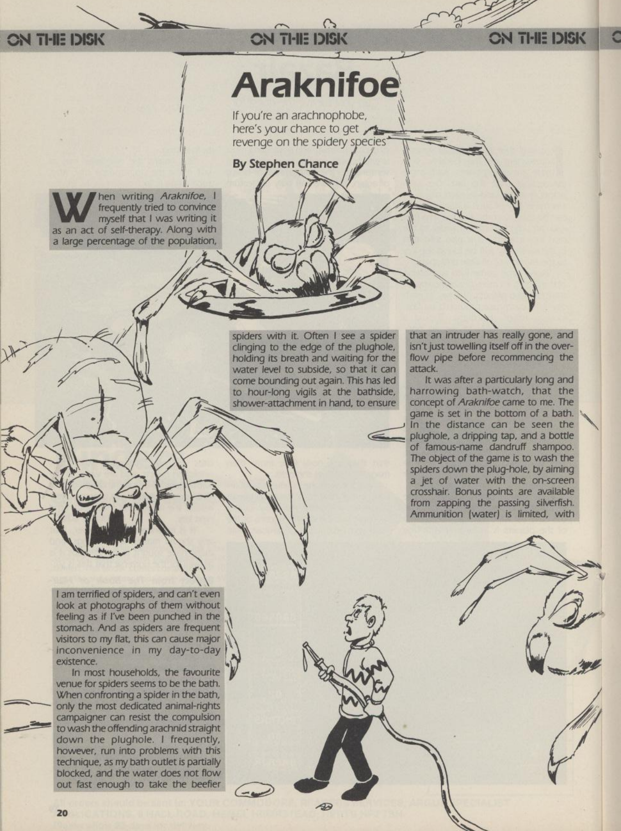
Loading the program

To load the program outside the menu, enter LOAD "BAZAIR",8 and RUN.

Araknifoe

If you're an arachnophobe, here's your chance to get revenge on the spidery species

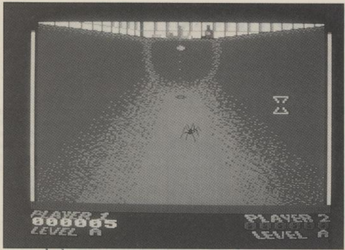
By Stephen Chance



When writing *Araknifoe*, I frequently tried to convince myself that I was writing it as an act of self-therapy. Along with a large percentage of the population, I am terrified of spiders, and can't even look at photographs of them without feeling as if I've been punched in the stomach. And as spiders are frequent visitors to my flat, this can cause major inconvenience in my day-to-day existence.

In most households, the favourite venue for spiders seems to be the bath. When confronting a spider in the bath, only the most dedicated animal-rights campaigner can resist the compulsion to wash the offending arachnid straight down the plughole. I frequently, however, run into problems with this technique, as my bath outlet is partially blocked, and the water does not flow out fast enough to take the beefier spiders with it. Often I see a spider clinging to the edge of the plughole, holding its breath and waiting for the water level to subside, so that it can come bounding out again. This has led to hour-long vigils at the bathside, shower-attachment in hand, to ensure that an intruder has really gone, and isn't just towelling itself off in the overflow pipe before recommencing the attack.

It was after a particularly long and harrowing bath-watch, that the concept of *Araknifoe* came to me. The game is set in the bottom of a bath. In the distance can be seen the plughole, a dripping tap, and a bottle of famous-name dandruff shampoo. The object of the game is to wash the spiders down the plug-hole, by aiming a jet of water with the on-screen crosshair. Bonus points are available from zapping the passing silverfish. Ammunition (water) is limited, with



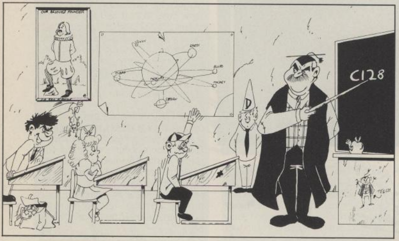
level indicators on either side of the screen. To refill, aim the crosshair over the drips from the tap and press the fire button. After clearing a bath, one travels to the next bath by a short burst of joystick waggling, while being chased by a particularly large spider.

The game is loaded by LOAD "ARAKNIFOE",8,1. When this main program file has loaded, which takes some time, RUN it and a shorter file called "AFOE2" will be loaded before the title screen is displayed. (The music is *In the Hall of the Mountain King* from *Peer Gynt* by Grieg.) The 1 or 2 player option is selected by pressing keys 1 or 2, and the game is started by pressing the fire button (joystick in port 2).

As I said, it seemed that writing a computer game about spiders might be a way to purge myself of my absurd feelings about them – designing 432 spider sprite definitions, spending months watching them scuttle across my monitor screen – but it wasn't to be: the spiders in my computer have turned out to be even more indestructible than the spiders in my bath.

Graphics Primer 128

Nick Gregory provides you with a suite of routines for the C128's 80-column mode



Graphics Printer loads into the area of memory usually occupied by the high resolution bit-map screen, from $1C00 to $3FFF. Not all of this space is used by the Primer (which is 2K long) so you can use the remaining memory to hold screen and/or character set data. The start of Ram not used by the Primer is 37 bytes above the start of the buffer. The address of the buffer you can get from Buffer (see below). You may think this is an odd place for it but the Primer was written for this location so that it can coexist with the object code generated by the Petspeed compiler and Oxford Pascal, both of which don't use this area of memory. If you are using interpreted Basic then you must reserve the memory using a command like:
GRAPHIC 1:GRAPHIC 0: GRAPHIC 5

Routine Structure

Each routine, and there are 31 in total, is accessed via a jump table located at the start of the Primer. To call a routine, a JSR instruction, or SYS in BASIC, is all that is needed once the appropriate parameters have been set up. The first two bytes of the jump table, however, are not part of this structure and are in fact the address (low byte/high byte) of the buffer used by the PRIMER. Most of the routines in the PRIMER use the buffer to store variables and constants while the routine is in operation, the contents of the buffer at any one time therefore are of no interest to the host program. The exception is when the host program is dealing with character data. Two routines, PTCHDA and GTCHDA, transfer character data to and from the buffer into the character RAM on the video controller. To exploit these routines the host program will need to know where the buffer is.

Routine Parameters

Some of the routines do not need parameters, HGR for example, simply puts the video controller into graphics mode. Other routines need one to four parameters and these are conveniently passed using the accumulator, X, Y registers and the status register. Still other routines require more than four parameters and to pass these the zero page locations $FA to $FE are used. One routine, PRTSTG, requires a call to INTSTG before it can be used effectively.

The Routines

The general principles I have just described 'sets the scene' so to speak for a detailed look at the routines which make up the Graphics Primer. I have listed the routines in the order which they appear in the jump table. Many of them you will already be familiar with, and so I will be giving only a detailed description where it is appropriate.

Name	BUFFER
Address	$1C02
Parameters	None
Notes	Not a routine but the address (low/high) of the buffer used by the primer. See above discussion on routine structure.

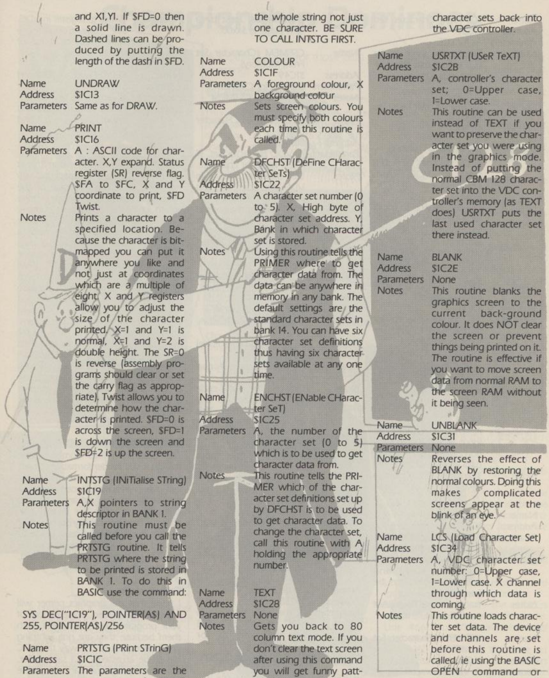
Name	HGR
Address	$1C04
Parameters	None
Notes	Sets up the high resolution graphics screen with 640 x 200 pixels. This routine also clears the graphics screen.

Name	CLRHGR
Address	$1C07
Parameters	None
Notes	Clears the graphics screen.

Name	PLOT
Address	$1C0A
Paramters	A and X registers, low/high byte of coordinate, Y register, Y coordinate.
Notes	Plots a point at X,Y where X is 0 to 639 and Y is 0 to 199.

Name	UNPLOT
Address	$1C0D
Parameters	Same as for PLOT.

Name	DRAW
Address	$1C10
Parameters	A,X and Y same as for PLOT. $FA,$FB and $FC as for A X and Y but destination coordinates. $FD set dash.
Notes	Draws a line between X,Y



and X1,Y1. If $FD=0 then a solid line is drawn. Dashed lines can be produced by putting the length of the dash in $FD.

Name UNDRAW
Address $1C13
Parameters Same as for DRAW.

Name PRINT
Address $1C16
Parameters A : ASCII code for character. X,Y expand. Status register (SR) reverse flag. $FA to $FC, X and Y coordinate to print, $FD Twist.
Notes Prints a character to a specified location. Because the character is bitmapped you can put it anywhere you like and not just at coordinates which are a multiple of eight. X and Y registers allow you to adjust the size of the character printed. X=1 and Y=1 is normal, X=1 and Y=2 is double height. The SR=0 is reverse (assembly programs should clear or set the carry flag as appropriate). Twist allows you to determine how the character is printed. $FD=0 is across the screen, $FD=1 is down the screen and $FD=2 is up the screen.

Name INTSTG (INiTialise STring)
Address $1C19
Parameters A,X pointers to string descriptor in BANK 1.
Notes This routine must be called before you call the PRTSTG routine. It tells PRTSTG where the string to be printed is stored in BANK 1. To do this in BASIC use the command:

```
SYS DEC("1C19"), POINTER(A$) AND
255, POINTER(A$)/256
```

Name PRTSTG (PRint STrinG)
Address $1C1C
Parameters The parameters are the same as those for PRINT.
Notes Prints a string to the graphic screen. Set up the parameters as for PRINT and this routine will print the whole string not just one character. BE SURE TO CALL INTSTG FIRST.

Name COLOUR
Address $1C1F
Parameters A foreground colour, X background colour
Notes Sets screen colours. You must specify both colours each time this routine is called.

Name DFCHST (DeFine CHaracter SeTs)
Address $1C22
Parameters A character set number (0 to 5). X, High byte of character set address. Y, Bank in which character set is stored.
Notes Using this routine tells the PRIMER where to get character data from. The data can be anywhere in memory in any bank. The default settings are the standard character sets in bank 14. You can have six character set definitions thus having six character sets available at any one time.

Name ENCHST (ENable CHaracter SeT)
Address $1C25
Parameters A, the number of the character set (0 to 5) which is to be used to get character data from.
Notes This routine tells the PRIMER which of the character set definitions set up by DFCHST is to be used to get character data. To change the character set, call this routine with A holding the appropriate number.

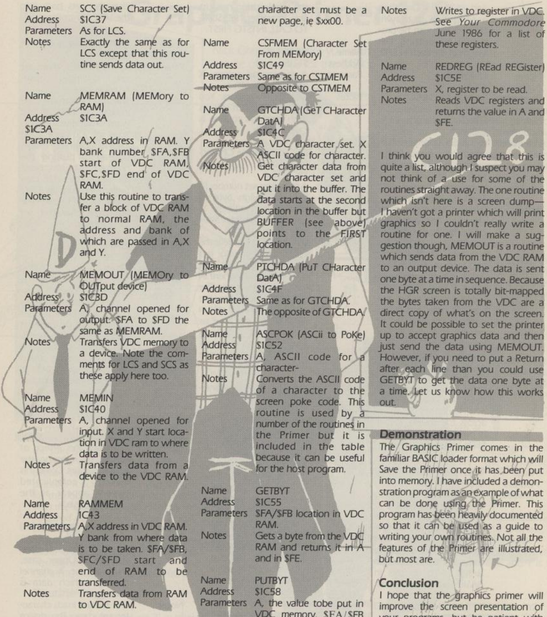
Name TEXT
Address $1C28
Parameters None
Notes Gets you back to 80 column text mode. If you don't clear the text screen after using this command you will get funny patterns because the data from the graphic commands will be interpreted as character data. This routine loads the normal character sets back into the VDC controller.

Name USRTXT (USeR TeXT)
Address $1C2B
Parameters A, controller's character set; 0=Upper case, 1=Lower case.
Notes This routine can be used instead of TEXT if you want to preserve the character set you were using in the graphics mode. Instead of putting the normal CBM 128 character set into the VDC controller's memory (as TEXT does) USRTXT puts the last used character set there instead.

Name BLANK
Address $1C2E
Parameters None
Notes This routine blanks the graphics screen to the current back-ground colour. It does NOT clear the screen or prevent things being printed on it. The routine is effective if you want to move screen data from normal RAM to the screen RAM without it being seen.

Name UNBLANK
Address $1C31
Parameters None
Notes Reverses the effect of BLANK by restoring the normal colours. Doing this makes complicated screens appear at the blink of an eye.

Name LCS (Load Character Set)
Address $1C34
Parameters A, VDC character set number: 0=Upper case, 1=Lower case. X channel through which data is coming.
Notes This routine loads character set data. The device and channels are set before this routine is called, ie using the BASIC OPEN command or assembler equivalent. The routine only moves the data so all other file parameters should already be established.



Name	SCS (Save Character Set)
Address	$1C37
Parameters	As for LCS.
Notes	Exactly the same as for LCS except that this routine sends data out.
Name	MEMRAM (MEMory to RAM)
Address	$1C3A
$1C3A	
Parameters	A,X address in RAM. Y bank number. $FA,$FB start of VDC RAM, $FC,$FD end of VDC RAM.
Notes	Use this routine to transfer a block of VDC RAM to normal RAM, the address and bank of which are passed in A,X and Y.
Name	MEMOUT (MEMOry to OUTput device)
Address	$1C3D
Parameters	A, channel opened for output. $FA to $FD the same as MEMRAM.
Notes	Transfers VDC memory to a device. Note the comments for LCS and SCS as these apply here too.
Name	MEMIN
Address	$1C40
Parameters	A, channel opened for input. X and Y start location in VDC ram to where data is to be written.
Notes	Transfers data from a device to the VDC RAM.
Name	RAMMEM
Address	1C43
Parameters	A,X address in VDC RAM. Y bank from where data is to be taken. $FA/$FB, $FC/$FD start and end of RAM to be transferred.
Notes	Transfers data from RAM to VDC RAM.
Name	CSTMEM (Character Set to MEMory)
Address	$1C46
Parameters	A VDC character set. X high byte of RAM address. Y Bank number.
Notes	Moves a character set from the VDC to RAM. The start address of the character set must be a new page, ie $xx00.
Name	CSFMEM (Character Set From MEMory)
Address	$1C49
Parameters	Same as for CSTMEM
Notes	Opposite to CSTMEM
Name	GTCHDA (GeT CHaracter DatA)
Address	$1C4C
Parameters	A VDC character set. X ASCII code for character.
Notes	Get character data from VDC character set and put it into the buffer. The data starts at the second location in the buffer but BUFFER (see above) points to the FIRST location.
Name	PTCHDA (PuT CHaracter DatA)
Address	$1C4F
Parameters	Same as for GTCHDA.
Notes	The opposite of GTCHDA.
Name	ASCPOK (ASCii to PoKe)
Address	$1C52
Parameters	A, ASCII code for a character-
Notes	Converts the ASCII code of a character to the screen poke code. This routine is used by a number of the routines in the Primer but it is included in the table because it can be useful for the host program.
Name	GETBYT
Address	$1C55
Parameters	$FA/$FB location in VDC RAM.
Notes	Gets a byte from the VDC RAM and returns it in A and in $FE.
Name	PUTBYT
Address	$1C58
Parameters	A, the value tobe put in VDC memory. $FA/$FB are the same as for GETBYT
Notes	The opposite of GETBYT.
Name	WRTREG (WRite to REGister)
Address	$1C5B
Parameters	A, value to be written. X register number.
Notes	Writes to register in VDC. See *Your Commodore* June 1986 for a list of these registers.
Name	REDREG (REad REGister)
Address	$1C5E
Parameters	X, register to be read.
Notes	Reads VDC registers and returns the value in A and $FE.

I think you would agree that this is quite a list, although I suspect you may not think of a use for some of the routines straight away. The one routine which isn't here is a screen dump—I haven't got a printer which will print graphics so I couldn't really write a routine for one. I will make a suggestion though, MEMOUT is a routine which sends data from the VDC RAM to an output device. The data is sent one byte at a time in sequence. Because the HGR screen is totally bit-mapped the bytes taken from the VDC are a direct copy of what's on the screen. It could be possible to set the printer up to accept graphics data and then just send the data using MEMOUT. However, if you need to put a Return after each line than you could use GETBYT to get the data one byte at a time. Let us know how this works out.

Demonstration

The Graphics Primer comes in the familiar BASIC loader format which will Save the Primer once it has been put into memory. I have included a demonstration program as an example of what can be done using the Primer. This program has been heavily documented so that it can be used as a guide to writing your own routines. Not all the features of the Primer are illustrated, but most are.

Conclusion

I hope that the graphics primer will improve the screen presentation of your programs, but be patient with them because they are not lightning fast (due mainly to the way the VDC has to be accessed). As I said earlier, assembler programs can use these routines unmodified as primitives in more complex procedures. If this happens perhaps you could share your routines with us through the pages of *Commodore Disk User*.

Championship Dominoes

While sitting in the comfort of your home you can have a game of dominoes with Max and Joe at one of the two levels available in this excellent domino game.

The game uses a double six set of dominoes and takes out all the drudgery of shuffing and starting the game as the computer checks who holds the highest double, or the highest value, if no doubles are held by any of the three players. Player three is the human contestant of course.

To make the game as realistic as possible redefined characters are used to display the dots of the dominoes including colours.

As the screen area available is limited certain changes to the game layout had to be made but these were carefully considered so as not to detract from the pleasure of the game. Only the last domino played at either end of the line is displayed, however, as a domino is played the value of the domino is added to the appropriate column on the right of the screen. This allows a check to be kept of all dominoes that have been played helping you to decide on your move.

The block at left of the screen with a number in it indicated the number of dominoes left in the pool which can be drawn if necessary.

How it works

The game consists of four parts which I will explain for those interested in programming.

Championship Dominoes

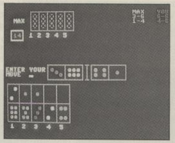
This is a short basic routine which when 'RUN' will 'LOAD' and 'RUN' the game automatically.

Champ1

A soothing game for those cold evenings

By Athol McEwan

This program is a machine code program which moves the screen memory to $CC00 (52224) and character memory to $C000 (49152) and moves the character ROM to this location to allow redefined characters to be used.



Champ2

This program has machine code programs for the following:

1. Displaying the first domino in your hand at $9510 (38166) and subsequent dominoes at $954C (38220).
2. Displaying the first centre domino at $96B6 (38582).
3. Displaying the centre domino to the left of the first centre domino at $9850 (38992).
4. Displaying the centre domino to the right of the first centre domino at $9863 (39011).
5. Displaying Max's first domino at $98EE (39150) and subsequent dominoes at $9911 (39185).
6. Displaying Joe's first domino at $992A (39210) and subsequent dominoes at $994D (39245).
7. Customised 'INPUT' routine at $9966 (39270). This routine will only react to digits, 'D', 'DEL' and 'RETURN' keys with a maximum of two characters. If the 'DEL' key is pressed the input is cleared and has to be entered from the beginning.

Champ3

This is the main Basic program for the game.

Line 110 to 380 sets up the game.

Line 380 to 430 deals the dominoes to the three players. To get a random deal three dominoes are randomly selected and then allocated to each player this is repeated until and selected number of dominoes are dealt.

Line 440 to 500 sets up the game screen.

Line 510 to 830 checks which player holds the highest double or highest value if no doubles are held.

Line 840 to 940 are the routines for playing the highest domino.

Line 950 to 1010 is a routine which decides who is next to play.

Line 1020 to 1130 allows you to enter your move and calls all the subroutines required. e.g. Remove a domino from your hand, draw an extra domino from the pool etc.

Line 1140 to 1640 is the routine which makes Max's move also calling subroutines required.

Line 1650 to 2100 is Joe's routine as for Max's move routine.

Line 2110 to 2360 is the subroutine which decides where to position the selected domino and calls the subroutine which prints the centre domino.

Line 2370 to 3690 are the subroutines.

Line 3700 to 3770 are the sound routines used.

Line 3780 to 4410 are the routines used for the end of the game options.

Line 4420 to 4650 are the variables which are set up before the first game.

Line 4660 to 5040 are the brief instructions for the game.

Line 5050 to 5180 are the three set of data used to redifine four characters which are used to create the centre dots fof your dominoes and the centre dominoes.

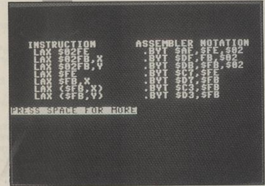
Loading the Program

To load the program outside the menu enter LOAD"DOMINOES",8 and RUN.

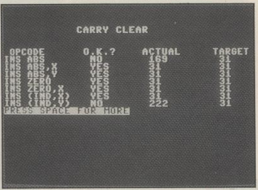
More Hidden Secrets of the 6510

Following on from "Hidden secrets of the 6510" in the July/August issue of CDU, this program will indicate which of the so-called "quasi op-code" instructions will work with your 64/128.

To load the opcode scanner outside the menu use LOAD "OPCODE",8. On running the program you will first be asked to set or clear the carry flag. This merely changes the machine code instruction before the instruction under test to SEC or CLC.



```
INSTRUCTION            ASSEMBLER NOTATION
 LAX $02FE              .BYT $AF,$FE,$02
 LAX $02FB,X            .BYT $DF,FB,$02
 LAX $02FB,Y            .BYT $DB,$FB,$02
 LAX $FE                .BYT $C7,$FE
 LAX $FB,X              .BYT $D7,$FB
 LAX ($FB,X)            .BYT $C3,$FB
 LAX ($FB,Y)            .BYT $D3,$FB
PRESS SPACE FOR MORE
```



```
              CARRY CLEAR

 OPCODE          O.K.?     ACTUAL       TARGET
INS ABS           NO        169           31
INS ABS,X         YES       31            31
INS ABS,Y         YES       31            31
INS ZERO          YES       31            31
INS ZERO,X        YES       31            31
INS (IND,X)       YES       31            31
INS (IND,Y)       NO        222           31
PRESS SPACE FOR MORE
```

To start with, press C to Clear the flag. Later on you will be able to restart the program and choose the other option. You will then be asked if you want to use a printer. First time round, use the screen option until you've solved the problem of possible crashes – explained later. All the printer routines use simple commands to increase the likelihood of them working on your printer. These routines were originally written for use with a DPS 1101.

You will then be presented with the program output. Reading across from left to right, the first column contains the addressing mode and the second whether or not it works. Thridly (fourthly too in the case of LAX) come the actual result or results and finally the target result.

LAX produces two results because the instruction loads the 'A' register and the 'X' register from a specified address, both these values needing to be checked against the target value.

The Menu

When you reach the end of the first part of the program, you will be presented with a menu. Option one, if selected, will list all the instructions which have worked. Option two will list all the instruction, together with the relevant .byt $xx,$xx,$xx form which must be used to represent these codes in an assembler.

Option three restarts the program. The arrow points to the last selection made at the set/clear carry prompt. Option four will reset the machine – you have been warned!

The machine code itself, filename "Oprg", loads from 49152 ($C000) to 49950 ($C31E) and is loaded automatically at the start of the program. The sys addresses for these routines can be found from lines 12000 plus in the basic program.

I have arranged, as far as possible, for the parameters used by the machine code routines to occupy particular locations. Also, I have tried to keep the input parameters themselves constant. They are usually $55 (85) (binary 01010101) amd 7 (binary 00000111).

The Unthinkable

While writing the program I was aware that if the processor didn't recognise a particular opcode, it might crash. The way around this is achieved by printing out a reference number which, if the machine doesn't lock-up, is overwritten by the data being displayed.

The reference number is in fact a line number. It refers to an SYS statement – a jump to the machine code routine for that instruction.

What to do

If the program does crash you will see the number on the left-hand side of the screen – write it down. Turn the computer off then on again and list the line number. Add a return statement to the beginning of that line.

Don't forget to do the same to other lines which may have caused a crash in previous attempts.

When you have found and corrected all these problems, save the program using SAVE"OPCODE1",8.

If you need to do this, don't forget to load Opcode1 instead of Opcode on future loads. Don't forget that if you are printing out the results, the computer may pause until the buffer empties. In this case the reference number will not appear.

Absolute addressing (absolute; absolute X and absolute Y) always uses $02FE (766) eg. ASO £02FE; ASO £02FB,X; INS $02FE (254) eg. LAX £FE; RRA £FB,X.

Indirect addressing uses £02FE indirected through $FE/$FF.

Some Preliminary Observations

I have run this program on a fairly old C64 and my own C128 (in 64 mode.) Surprisingly the results were identical! Certain anomalies have, however, appeared on both machines.

The first concerns RLA or "Rotate Left then AND result with accumulator". This instruction is like half a rotate – ie the Most Significant Bit of the byte concerned is lost and the carry bit is moved into the Last Significant Bit.

The same is true for RRA except that the LSB is lost and the carry is moved into the MSB. INS is also affected by the state of the carry flag. The increment part of the instruction is carried out ONLY if the flag is set.

It should be noted that different target results are used for RLA, INS and RRA when the carry is set or cleared. Some of the instructions do actually appear to operate but not in the way described. I hope to look into these and hope I've generated enough interest for you to do the same!

Your Turn

The ultimate aim of this program is to discover how many machines these codes will work on. I suspect that many will have the same effect on all machines. To help us coders in the future, it would be helpful if as many people as possible sent in the results this program produces on their machine to me via CDU for correlation, the results to be printed at a later date. If you have a printer, a full printout would be appreciated. If not, a list of those codes which work with the carry clear would be just as welcome.

One important thing to remember is that codes which caused a crash may be shown as working in your results because in this case the processing routines would use the values from the last code worked on. Don't forget to remove them from your results.

INSTRUCTION	ASSEMBLER NOTATION
LAX ABS	•BYT $AF,$FE,$02
LAX ABS,X	•BYT $DF,$FB,$02
LAX ABS,Y	•BYT $DB,$FB,$02
LAX ZERO	•BYT $C7,$FE
LAX ZERO,X	•BYT $D7,$FB
LAX (IND,X)	•BYT $C3,$FB
LAX (IND,Y)	•BYT $D3,$FB
ASO ABS	•BYT $OF,$FE,$02
ASO ABS,X	•BYT $IF,$FB,$02
ASO ABS,Y	•BYT $IB,$FB,$02
ASO ZERO	•BYT $07,$FE
ASO ZERO,X	•BYT $17,$FE
ASO (IND,X)	•BYT $03,$FB
ASO (IND,Y)	•BYT $0B,$FB
RLA ABS	•BYT $2F,$FE,$02
RLA ABS,X	•BYT $3F,$FB,$02
RLA ABS,Y	•BYT $3B,$FB,$02
RLA ZERO	•BYT $27,$FE
RLA (IND,X)	•BYT $37,$FB
RLA (IND,Y)	•BYT $23,$FB
RLA IMM	•BYT $2B,$55
INS ABS	•BYT $EF,$FE,$02
INS ABS,X	•BYT $FF,$FB,$02
INS ABS,Y	•BYT $FB,$FB,$02
INS ZERO	•BYT $E7,$FE
INS ZERO,X	•BYT $F7,$FB
INS (IND,X)	•BYT $E3,$FB
INS (IND,Y)	•BYT $F3,$FB
LSE ABS	•BYT $4F,$FE,$02
LSE ABS,X	•BYT $5F,$FB,$02
LSE ABS,Y	•BYT $5B,$FB,$02
LSE ZERO	•BYT $47,$FE
LSE ZERO,X	•BYT $57,$FB
LSE (IND,X)	•BYT $43,$FB
LSE (IND,Y)	•BYT $53,$FB
RRA ABS	•BYT $6F,$FE,$02
RRA ABS,X	•BYT $7F,$FB,$02
RRA ABS,Y	•BYT $7B,$FB,$02
RRA ZERO	•BYT $67,$FE
RRA ZERO,X	•BYT $77,$FB
RRA (IND,X)	•BYT $63,$FB
RRA (IND,Y)	•BYT $73,$FB
AXS ABS	•BYT $8F,$FE,$02
AXS ZERO	•BYT $87,$FE
AXS ZERO,X	•BYT $97,$FB
AXS (IND,X)	•BYT $83,$FB
STZ ABS	•BYT $9C,$FE,$02
ALR IMM	•BYT $4B,$07
SAX IMM	•BYT $CB,$20
TAD ABS	•BYT $8B,$20

OPERATIVE OPCODES
(CARRY CLEAR)
LAX ABS
ASO ABS
ASO ABS,X
ASO ABS,Y
ASO ZERO
ASO ZERO,X
ASO (IND,X)
RLA ABS
RLA ABS,X
RLA ABS,Y
RLA ZERO
RLA ZERO,X
RLA (IND,X)
INS ABS,X
INS ABS,Y
INS ZERO
INS ZERO,X
INS (IND,X)
LSE ABS
LSE ABS,X
LSE ABS,Y
LSE ZERO
LSE ZERO,X
LSE (IND,X)
RRA ABS
RRA ABS,X
RRA ABS,Y
RRA ZERO
RRA ZERO,X
RRA (IND,X)
AXS ABS
AXS ZERO
AXS ZERO,Y
AXS (IND,X)
ALR IMM
SAX IMM
TAD ABS

High Speed Graphics

In this issue's installment of our continuing series, Allen Webb adds spice to your sprites

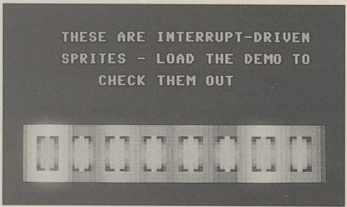
This month I will cover the handling of sprites. A sprite is a user-definable pattern which can be moved to any position on the screen. Their use allows a wide range of animation effects for use in games. The main irritation is that their use is fiddly and slow. The problem is that sprites are controlled by individual bits in a set of registers in the VIC chip. Tweaking these from BASIC involves a number of POKE/PEEK statements with the use of AND and OR operations. Overall this leads to sluggish animation. In this section I will provide a set of machine code routines which will ease access to the sprites and will give a slight increase in running speed.

First some background information. A sprite can obtain its design from almost anywhere in the memory in the current video bank. This design occupies 64 bytes and all you need to do is tell the system which block of data to use for each sprite. Since the VIC chip can access only 16K at a time, the maximum number of patterns available is 16384/64 or 256. Pattern number 0 occupies the memory addresses from 0 to 63, pattern 1 occupies memory addresses from 64 to 127 and so on. Most of you will use a sprite designer which will take care of the SAVEing and LOADing of the design data. In part 1 of this series, I gave details of the memory map used. The number of sprite designs available to you will depend on the number of redesigned character sets you use. Here is a repeat summary.

Number of Redefined Character Sets	Pattern Numbers Available	Number of Sprites Available
1	160-255	95
2	192-255	63

Each sprite has a number of attributes which you can manipulate:

1. Colour. As in the case of characters, sprites can have two colour modes:
*High resolution. This uses a single colour and each sprite can be assigned its own individual colour.
*Multicolour. This allows sprites to have four colours. Three colours are fixed for all sprites and one is sprite specific.
2. Size. You can have sprites in two sizes in either of the vertical or horizontal directions.



```
THESE ARE INTERRUPT-DRIVEN
SPRITES - LOAD THE DEMO TO
   CHECK THEM OUT
```

3. Positions. You can specify the vertical and horizontal positions of the sprites on the screen. If X is the horizontal position and Y is the vertical position, the visible portion of the screen is defined by (unexpanded sprites):

0 X 344
30 Y 249

4. Priority. Sprites can be behind or in front of the text on the screen.

The code occupies the memory from $1800 to $1A55. The routines are called from a jump table starting at $1800 so all you need is to specify the offset from this address. The routines are summarised in table 1:

The following detailed descriptions of the routines assume that SA has been defined with the value 6144. As usual, no error checking is included in routines. You must therefore ensure that your program does not use illegal values. Most importantly, the sprite number must not be outside the range 0 to 7. Any other oddities will be described below.

Table 1

Offset of routine		
1	0	Function of routine
2	3	Set sprite pattern
3	6	Turn off sprite
4	9	Set sprite colour
5	12	Horizontal expand
6	15	Vertical expand
7	18	Set colour mode
8	21	Set several attributes
9	24	Position sprite
10	27	Turn on interrupts
11	30	Turn off interrupts
12	33	Enable sprite animation
13	36	Disable sprite animation
14	39	Set sprite priority
15	42	Set colour registers

Routine 1

Jump Table Offset: 0
Function: This routine sets the specified sprite to the specified sprite pattern.
Syntax: SYS SA, SP, PATTERN NO
Example: SYS SA,1, 192
Parameters ranges: 0 =SP =7, 0 = PATTERN NO =255

Routine 2

Jump Table Offset: 3
Function: This routine turns the specified sprite on.
Syntax: SYS SA+3, SP
Example: SYS SA+3,1
Parameter ranges: 0 = SP =7

Routine 3

Jump Table Offset: 6
Function: This routine turns the specified sprite off.
Syntax: SYS SA+6, SP
Example: SYS SA+6,1
Parameter ranges: 0 = SP =7

Routine 4

Jump Table Offset: 9
Function: This routine sets the specified sprite to a specified colour.
Syntax: SYS SA+9, SP, COLOUR
Example: SYS SA+9,1,7
Parameter ranges: 0 = SP =7, 0 = COLOUR =15

Routine 5

Jump Table Offset: 12
Function: This routine toggles the horizontal expansion of the specified sprite.
Syntax: SYS SA+12, SP, FLAG
Exampale: SYS SA+12,1,0
Parameter ranges: 0 =SP =7. FLAG = 0, unexpanded sprite. FLAG 0, expanded sprite.

Routine 6

Jump Table Offset: 15
Function: This routine toggles the vertical expansion of the specified sprite.
Syntax: SYS SA+15, SP, FLAG
Example: SYS SA+15,1,0
Parameter ranges: 0 =SP =7. FLAG=0, unexpanded sprite. FLAG 0, expanded sprite.

Routine 7

Jump Table Offset: 18
Function: This routine sets the colour mode of the specified sprite.
Syntax: SYS SA+18, SP,FLAG
Example: SYS SA+18,1,0
Parameter ranges: 0 =SP =7. FLAG=0, high-resolution sprite. FLAG 0, multicolour sprite.

Routine 8

Jump Table Offset: 21
Function: This routine sets up a number of parameters simultaneously.
Syntax: SYS SA+21, SP, COLOURMODE, XEXPAND, YEXPAND, COLOUR
Example: SYS SA+21,1,1,0,0,7
Paramater ranges: 0 =SP =7. The other flags have the same values and effects as described for routines 7, 5, 6 and 4 respectively.

Routine 9

Jump Table Offset: 24
Function: This routine positions the specified sprite.
Syntax: SYS SA+24, SP, XPOS, YPOS
Example: SYS SA+24,1,100,66
Parameter ranges: 0 =SP =7. The visible limits of sprite positions were given earlier. The use of values of YPOS greater than 256 gives wrap around on the screen.
Values of X in excess of the screen width simply gives an unpredictable sprite position. Whilst both of these situations give an odd effect, they will not crash the machine.

Routine 10

Jump Table Offset: 27
Function: THis routine turns the animation interrupts on.
Syntax: SYS SA+27

Routine 11

Jump Table Offset: 30
Function: This routine turns the animation interrupts off.
Syntax: SYS SA+30

Routine 12

Jump Table Offset: 33
Function: This routine sets the animation parameters of the specified sprite and starts the animation sequence.
Syntax: SYS SA+33, SP, PATTERN1, PATTERN2, DELAY
Example: SYS SA+33,1,190,198,20
Parameter ranges: 0 =SP =7. 0 = PATTERN 1 =255, =PATTERN2 =255, 0 =DELAY =255

Routine 13

Jump Table Offset: 36
Function: This routine stops the animation of the specified sprite.
Syntax: SYS SA+36, SP
Example: SYS SA+36,3
Parameter ranges: 0 =SP =7.

Routine 14

Jump Table Offset: 39
Function: This routine sets the priority of the specified sprite.
Syntax: SYS SA+39, SP, FLAG
Example: SYS SA+39,1,1
Parameter ranges: 0 =SP =7. FLAG = 0 puts the sprite in front of the screen contents. FLAG 0 puts it behind.

Routine 15

Jump Table Offset: 42
Function: This routine sets up the relevant colour registers.
Syntax: SYS SA+42, BORDER, SCREEN, SPRITECOLOUR1, SPRITECOLOUR2
Example: SYS SA+42, 1,2,3,4
Parameter ranges: All parameters are the same: 0 =COLOUR =15

The animation routine needs a little expansion. Once set up the interrupts act as a background task. It is called every 60th of a second and it scans to see which sprites are to be animated. Each sprite has a flag which decides whether it is animated. Routine 12 decides a number of values. The animation cycles through a sequence of designs which MUST be continuous within the memory. You must specify the number of the starting pattern (PATTERN1) and the finishing pattern (PATTERN2). Clearly the start pattern must have a lower number than the end pattern. If not, the sprite will cycle through all 256 possible sprite patterns. DELAY determines how quickly you step through the sequence in 60ths of a second. A value of 30 will change pattern every 30/60 or 0.5 seconds. A value of 1 will change the pattern every 1/60 second. Due to the way the routine works, a value of 0 gives the longest delay of almost 4.5 seconds per pattern.

Two important points must be made:

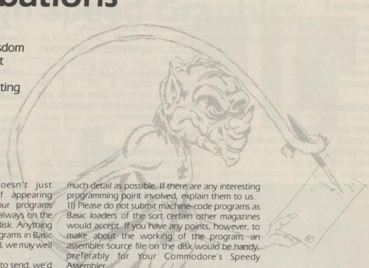
*The animation routine will not operate with the raster routine given in part 1. This is regretted but this due to the fact that the routines have been written individually. In any event, the operation of too many interrupts can slow down BASIC and have other odd effects.

*Always disable the interrupts and turn off sprites if you plan to access the disk drive.

As before you must raise the bottom of BASIC before you can use the routines. The demo on the disk should give you some idea of how to use the routines. ■

Contributions

Written some programs? Got some programming wisdom to pass on? Or do you want to write about your own fields of interest? We're waiting for your contributions.



Commodore Disk User doesn't just offer you the chance of appearing in print, but of putting your programs on our disk for all to admire. We're always on the lookout for new programs for the disk. Anything goes, utilities, games or business programs in Basic or machine code - if we think it's good, we may well publish it.

Even if you haven't got a program to send, we'd love to pick your brains. If you have a field of expertise you'd like to explain or any tips and hints of interest to disk users, send them in.

But how do you go about preparing a submission? Just follow the guidelines and all should go well. You don't have to be a great novelist to contribute, but if you follow our simple rules then it will make our job a lot easier.

1) If possible all material sent to the magazine should be typed or printed out on a computer printer.

2) All text should be double-spaced, i.e. there should be a blank line between each line of text. You should also leave a margin of at least 10 characters on each side of the text.

3) On the first page you should put the following:

Name of the article
Machine that it is for (C64/128)
Any extras required - disk, printer, add-ons etc.
Your name
Your address
.'our telephone number

4) The top of every page should have the following information on it:

Abbreviation of the article title
Your name
The page number

For example, suppose you had submitted a piece on C64 3D graphics. You should put something like this at the head of the page:

3D/G. Brown/1

5 Please make sure that you do not make any additional marks on your text, especially underlining.

6) Try to write in clear concise English. Your contribution does not have to be a great work of literature, but it must be comprehensible.

7) On the bottom of each page you should put the word MORE if there are more pages to the article, or ENDS if it is the last page.

8) If possible, enclose a listing of all programs.

9) Use a paperclip to hold the pages together. Do not staple them.

10) When submitting programs for the disk, submitting the program alone is not enough. Please tell us how to load, run and use it, preferably in as much detail as possible. If there are any interesting programming point involved, explain them to us.

11) Please do not submit machine-code programs as Basic loaders of the sort certain other magazines would accept. If you have any points, however, to make about the working of the program, an assembler source file on the disk would be handy, preferably for Your Commodore's Speedy Assembler.

12) Programs for the disk should be in as few chunks as possible. This makes our disk menu easier to set up.

13) Programs under 10 lines can be included in the text. If your program is longer than this it must be on a disk.

14) If your article needs any artwork, then supply clear examples of what you want. We don't expect you to be an artist, but we do need to see what is required.

15) Photos, if necessary, must be either black and white prints or colour slides. We can take shots ourselves, so don't worry about this too much.

16) Submissions of any length are welcome. A five-line routine may be just as welcome as a six-part series of 2000-word articles.

17) Payment can vary from £50 for a very short routine to £700 for a large program published in installments, and depends on quite a number of factors, such as complexity and presentation of program. For articles, the number of magazine pages taken up is the salient factor.

18) All payments are made in the month that the magazine containing your article has appeared in print.

19) If we do find your submission suitable for inclusion in the magazine, we will write to you giving the terms of publication, the rate of payment, and an agreement form. Prompt return of this form will allow us to use your program as soon as possible.

20) If you want the program to be returned to you, should we find it suitable for publication, then you should enclose a stamped addressed envelope.

21) If you use a wordprocessor, then enclose a copy of your text on the disk and state clearly which wordprocessor you use.

22) Send your programs and articles to:

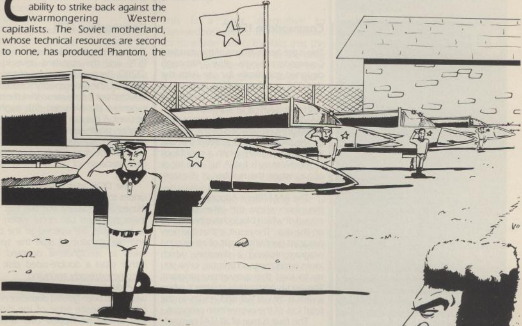
Commodore Disk User
Submissions
1 Golden Square
London W1R 3AB

23) Commodore Disk User cannot accept any liability for items sent to the magazine.

Phantom

Strike a blow for international revolution with this fast action game

Comrade! At last we have the ability to strike back against the warmongering Western capitalists. The Soviet motherland, whose technical resources are second to none, has produced Phantom, the ultimate in multi-role, all-wether, stealth-capable, high-precision, high-lethality attack technology.

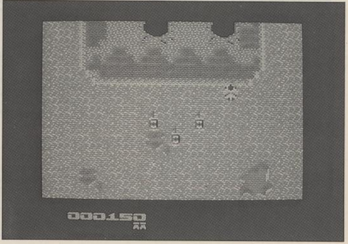


Phantom is a Mach 5 interceptor/ fighter bomber capable of forming the mainstay of our heroic Soviet defence system. Of course, tovarische, it is necessary for all Phantom pilots to practice offensive maneouvres – the Politbureau is determined that no future conflict will be fought on the sacred soil of the USSR.

To this end all heroic Soviet pilots are required to fall in for simulation training covering a possible Phantom ground support/interdiction mission over Nato territory. The simulated targets will vary between a range of enemy ground forces to selected air and sea targets.

Today, comrade, you will face a drill – even this will not be easy. Tomorrow – who knows – the future may be in your hands. The capitalists must pay a price in blood for every square inch of our soil!

Phantom was produced using the Shoot-Em-Up Construction Kit from Palace Software. Control is via the joystick. To load the game outside the usual menu, enter LOAD "PHANTOM",8 and RUN. ■



Six Drives

How can you choose a drive now that the 1541 monopoly has been broken?

By Kerry Fowler

Disk drives are as expensive as a computer and it's important to choose the right one to suit your present and future needs. With six drives to choose from, this may be no easy task.

Of the six drives, four are Commodore devices: the 1570, 1571, 1541C and 1581 drives. The other two are the Oceanic and the Blue Chip drives, which raises the obvious question of compatibility. It is difficult to say for sure that the two drives are totally compatible especially given that the 1541C is not 100 per cent compatible with some of the commercial disks produced especially for the earlier 1541 drive!

Such cases are thankfully rare and my experience is that I have never had any problems trying to load software on any of the drives mentioned in this article. If you have ever experienced any incompatibility problems, we'd be interested to hear about them and then our investigative team will look into the situation and report back through the pages of *Commodore Disk User*.

Realising that I'm preaching to the converted because the majority of readers will already have purchased a drive (we hope!), I will take time out to explain why a second drive can be useful and suggest a few questions that you should ask yourself.

A second drive is extremely useful when making backup disks. A full disk backup means several interchanges of the source and target disks and this constant disk swapping can become a chore. Similarly, when a game uses several disks the constant insertion and removal of disks increases the possibility of the disk surface being scratched as well as exposing these often expensive programs to unnecessary wear and tear.

Before buying a second drive, weigh up the pros and cons. Will the cost be offset by the amount of use? Do the drives have to be the same type? Can the drive numbers be altered easily? Is the new drive purely a backup or will it get used?

Commodore 1541C

This is the latest version of the standard Commodore single drive. Although designed specifically for use with the C64/128 machines, it does have a few nasty little bugs. The major problem is with the save and replace function (SAVE" 0:...). I've yet to see a fully satisfactory explanation of the reasons for the problem but it appears that the BAM goes haywire when a data block in the directory is full. A full data block occurs when the number of directory entries is a multiple of eight.

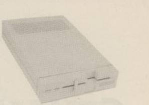
The BAM (Block Allocation Map) is the place where the drive keeps its record of which blocks have been used on the disk. The result is that the new program overwrites part of an existing program, leaving a directory which looks deceptively acceptable, until you try to load the overwritten program. This bug is particularly apparent when a disk is almost full and results in the total loss of the overwritten program.

This fault is true of all 1541 compatible Commodore drives from the old 4040 to the latest 1571.

The advantage of using a Commodore 1541 drive is that all disk turbos can be used, including hardware systems such as Dolphin DOS.

Commodore 1570

This was the C128 development of the 1541 disk operating system (DOS) and includes a faster data movement system which can be utilised with the 128 mode whether in Basic or CP/M. The system was soon superceded by the 1571 drive.



Commodore 1571

This is a double-sided version of the 1570 and the standard drive in the C128D. Because both sides of the disk are used, the disk capacity is effectively doubled but it does mean that the more expensive double-sided disk ought to be used. A risk can be taken with good quality single-sided disks but these carry no guarantee. The normally unused side may not be able to take data because of an unsound oxide coating.

The DOS adds a few more peculiarities to the system. Most can be dismissed as irritations but the fact that major problems can occur when two or more files are opened to the drive and the amount of time the system takes to recognise a doubled disk rather than a double-sided disk can become extremely annoying.

A doubled disk is a single sided disk which has been given a duplicate notch on the opposite edge of the disk so that it can be flipped over and re-inserted for use in a 1541 drive. Known colloquially as a flippy, this system is employed by many companies which produce multidisk packages so that costs can be kept to a minimum. When the 1571 is faced with such a disk, it isn't expecting a format of this type and the operating system remains confused for a while.

The solution is expensive but can be obtained from Financial Systems Software in the form of the 1571 FIX ROM for £24.95.

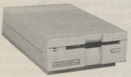
The great advantage with the 1571 drive is its ability, in 128 mode, to mimic other systems' formatting types. This means that is is possible to convert Commodore files on to a MS-DOS compatible disk, or vice versa! I have used the Big Blue Reader program for this and it works perfectly well, giving a useful cross over between the IBM PC and C128 without the need to own a PC or to use RS232 linking. As with the FIX ROM, the Big Blue Reader is available from Financial Systems Software for £34.95.

Commodore 1581

This is the surprising new addition to the Commodore range which breaks with the conventional 5.25-inch disk system and replaces it with a 3.5-inch disk system which is the same type of disk that the Amiga uses.

Although dedicated software is thin on the ground, this can make an excellent second drive for those who need to store massive databases or who want to pack as many programs into as small a storage space as possible.

Unlike the normal disk, the 3.5-inch disk can store up to 3 160 blocks of information. This means over 700 kilobytes of storage space opposed to less than 160 kilobytes on a standard single-sided floppy.



Oceanic OC-188N

Various forms of this drive have appeared as the Excelerator series from Evesham Micros. Now, the British name has been dropped but the drive remains the same.

This is the smallest 1541 compatible drive and possibly the most attractive from the price point of view. The price is the lowest for any of the drives and the basic GEOS system software is included in the price.

The drive covers an area which is only fractionally larger than two disks and its height is about half that of the 1541. Beneath the drive there is a bank of switches for changing the device number and offers the simplest method of any of the other drives.

Blue Chip

This is a neat alternative to Commodore's drives. The main reason for its smaller size is the fact that the power supply is external to the drive and can be tucked away on the floor to relieve the pressure on desk space that is normally suffered by micro owners.

Internally, the PCB boards and drive mechanism closely resembles the 1541 but it is a different, though compatible, machine and may not work with all disk turbos. In tests, it couldn't be faulted but these tests, though rigorous, could not be classed as fully exhaustive.

Unfortunately, one thing that is shared with the Commodore machines is the inaccessible device number changer. On the Commodore machines, hard-wiring a device number means breaking links on the PCB's silver tracery of circuitry. For the amateur, this is off-putting and difficult to reverse. On the Blue Chip, the device number is changed by cutting wires. This may be slightly less daunting but it is also easier to reverse, if necessary.

Conclusions

As a first drive I would tend to be conservative and suggest the Commodore 1541 for C64 owners and 1571 for 128 owners. In the case of the C128, this may be a forced option with the C128D supplies outstripping the dwindling stocks of the standard C128. By making this decision, any spurious incompatibility problems can be directly thrown back at Commodore or the software house concerned.

As a second drive, any of the machines compatible with your computer is a reasonable choice. My personal preference must be the Oceanic drive, both because of its price and the fact that it comes with a useful multiple device system, the GEOS package.

Disk Drive Comparison Table

COMPUTER	MODEL	SIZE (mm) L B D	SUPPLIER*	PRICE	COMMENTS
C128	1570/71	370x200x100	FSSL 18 High Street Pershore Worcs WR10 1BG	£184.95	Standard with 128D. Has ROM vagaries but assures total compatibility. Some problems with turbos in 1541 mode.
C128/C64	1581	230x140x65	FSSL (see above)	£184.95	Larger storage capacity. Uses 3.5 inch disks. Higher transfer speeds. Not very many dedicated packages.
C64	1541C	370x200x100	Delta Pi Software 8 Ruswarp Lane Whitby N Yorks YO2 1ND	£159.00	The drive that most of the disk utilities and alignment kits are written for.
C64	Oceanic	275x150x50	Dimension Computers 27/29 High St Leicester LE1 4FP	£129.95	Supplied with GEOS and one of two software packs. Small with accessible device number switches.
C64	Blue Chip	290x170x75	Datel Electronics Fenton Ind Est Govan Road Fenton Stoke-on-Trent	.£139.99	Sturdy drive with stylish appearance.

*Best published prices at time of going to press.

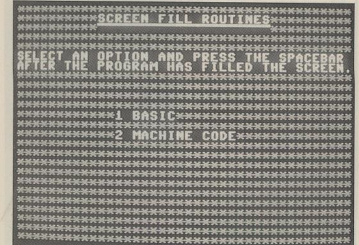
Deciphering Code

Machine code makes the brave tremble but it can be as easy as Basic

By Eric Doyle

It has often been written that machine code is much faster than Basic but one question that isn't often answered is 'Why?'. After all, Basic is written in machine code so there shouldn't be that much difference.

The answer to the speed difference lies in the interpreter. The program has to handle line numbers, jumps, keywords, functions within functions and parameters of all kinds. In addition all errors must be tested for, often several times, and all cases catered for. Plain machine code on the other hand has all of its information fed in directly and the only errors it checks for are those that the programmer chooses.



```
SCREEN FILL ROUTINES

SELECT AN OPTION AND PRESS THE SPACEBAR
AFTER THE PROGRAM HAS FILLED THE SCREEN.

1 BASIC
2 MACHINE CODE
```

There are two demonstrations in the program on this month's disk, one which fills the screen from Basic and one which uses the same method but in code. The speed difference is like comparing a snail to a Jaguar.

The great drawback is that machine code is so much simpler than Basic. This may sound surprising but it's actually true and it's this simplicity that makes code so much harder to use!

Enough of these enigmatic statements, let me explain. Code can only POKE and PEEK locations, add and subtract, perform logical operations, or perform GOTOs and GOSUBs. To get the most from code, the programmer needs a thorough understanding of how the computer's memory works. This can take weeks, months or years depending on the programmer's ability. Some concepts are obvious and equate to tasks in Basic, others are a lot trickier.

Into the Heart

The operations centre of the computer is the 6510 microprocessor. This is a complex device of which I know very little because most of what it does is totally irrelevant to the programmer. It's rather like a sausage machine – as long as you know how to feed suitable material in one end, you know what will come out at the other end.

The heart of the CPU is the *accumulator*. This is the register that holds the number to be acted upon and if you ignore its high-sounding title and think of it as a special memory location into which values can be poked, you won't be far from the mark.

To put a number into the accumulator you feed two numbers in succession into the CPU. First of all comes a code number which tells the CPU to load the second value into the accumulator. The code number is 169 in decimal, don't worry about hexadecimal and binary because they're irrelevant at this point. The number which follows can be any number from 0 to 255, for this example we'll use 42 – the screen code for an asterisk. To then store this value in a memory location such as the screen another special code is used, 141, followed by the memory location 1024.

The problem is that the CPU can only cope with numbers less than 255 so how can 1024 be specified? This is where the high (hi-byte) and low byte (lo-byte) principle comes in. For now just accept that any number from 256 to 65535 can be specified as a two byte number by the following method. First divide the number by 256 to give the hi-byte as the part of the number in front of the decimal place. If the number is an exact whole number then all is well because the lo-byte is zero.

More often than not there will be a decimal part to the number. The decimal is the lo-byte divided by 256. It must be returned to a whole number. Although this could be done by multiplying it by 256, the vagaries of significant decimal places may give a number with a few decimal places after it. The best solution is to multiply the hi-byte by 256 and then subtract it from the original figure. In Basic, calculating the hi-byte would be represented by:

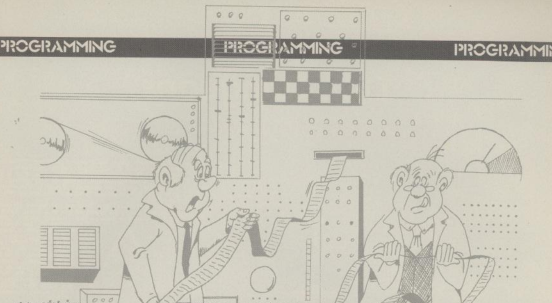
HI=INT(NUMBER/256)

and the lo-byte would then be worked out from this:

LO=NUMBER-(HI*256)

A good exercise for any doubting Thomas would be to take several values from 256 to 65535 and substitute them in the equations for the variable NUMBER. Whichever value is chosen the values of HI and LO will never exceed 255.

Applying this rule to 1024 will give a HI value of four and a LO value of zero. So the command number 141 tells the CPU to expect two numbers following it to form an address. The numbers are arranged in lo-byte/hi-byte order which is merely a convention



of the 6510 chip.

We now have a program:

```
169, 42, 141, 0, 4
```

Ardent hackers may recognise this as the sort of DATA statement that is found in so-called Basic loaders that mystically produce machine code programs.

The simple routine will be entered from Basic by using SYS. This is the Basic codeword which causes a program to jump from Basic's control and into a machine code program. The command needs a value after it and this is the location in memory where the first command of the machine code lies. In our example this is 169. A suitable place for machine code is the area from 49152 to 53247 but how do we get the numbers there?

Back to Basic

For the moment the numbers will have to be poked into position and, since it isn't a complex program, you'll have to type this in for yourself.

If we did this at the moment the code would put a screen poke value for an asterisk into the accumulator and then transfer it to the screen. Then what? No, the program wouldn't automatically return to Basic, the computer would simply hang up with an asterisk at the top left-hand corner of the screen! To get control back to Basic we need another command for the code which is the number 96. Our program becomes:

```
10 FOR A =49152 TO 49157
20 READ V
30 POKE A, V
40 NEXT
50 DATA 169, 42, 141, 0, 4, 96
```

After running the program, press the shift key and CLR/HOME to clear the screen and then type SYS 49152. The S of SYS should change to an asterisk and you've written your first machine code program.

Code Monitoring

Two words which confuse fledgling coders are *monitors* and *assemblers*. Machine code is also sometimes referred to as assembler, or assembly, code. As you will see, the correct name is machine code when referring to the raw numbers, so where do these strange words creep in?

When a machine code routine is called by a SYS command, the compouter assumes that the first number it encounters will be a command value (such as the 169 in our program). This then tells the CPU that the next value is a number to be acted upon so the CPU obeys. It then has nothing to do so it takes the next number as a command and acts accordingly. It is therefore possible to have code which starts 169, 169 and the CPU will translate this as meaning 'load the value 169 into the accumulator'. If the SYS command accidentally points to the second 169, the CPU will try to load the number which follows it into the accumulator and it won't be long before the machine crashes.

Consequently, these *action* numbers are like Basic keywords. If the computer reads in the word POKE it then assumes that the next value is a memory location which will be followed by a comma and then a value which is to be stored in this location. If these is not so, a SYNTAX ERROR will be printed on the screen. Machine code is not so generous, if the code doesn't make sense the computer will crash out or hang up and can only be reset by switching off and trying again.

How can a mere human decipher a list of numbers? Take this secion of code as an example:

```
169, 169, 169, 141, 141, 141....
```

Without starting at the beginning of the program and working through, it's impossible to say for sure what the program is doing. The code could be at then end of a 4K block of code which would take forever to decipher. There must be a better way.

To help the programmer to understand the code, the action numbers have been give names. For example, 169 is known as LDA which is a mnemonic for *LoaD* the Accumulator and 141 is STA or STore the Accumulator. So to rewrite our code in this new language it becomes:

```
LDA 42
STA 00, 04
RTS
```

Can you guess what RTS stands for? It returns from code to Basic in this case and means ReTurn from Subroutine. In other words what we've written is a kind of subroutine but in pure code rather than in Basic.

Deciphering the code as mnemonics is known as assembler code but, although it helps the programmer, it means nothing to the programmer. So a translator is needed. This is the monitor but before we learn more we'll first have to look at hexadecimal in more detail.

In the meantime try different values in the example program for 42 and calculate new high and low bytes for screen locations (1024 to 2023) and substitute them for 00, 04.

Disk Dungeons

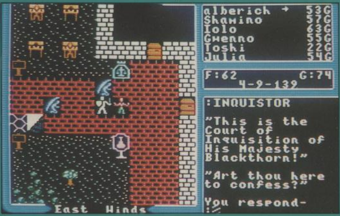
Gordon Hamlett comes up with more news from the adventure caverns.

With the continued proliferation of computer role playing games, we offer some guidance for would be heroes.

Ultima V

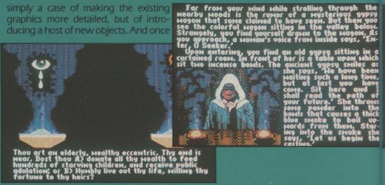
At last! Over a year since it was first announced, *Ultima V* has finally arrived on the 64. The Ultima series has always been first among role playing systems, not becuase of pretty graphics or anything like that, but because the author, Lord British, also known as Richard Garriott has paid most attention to the one detail that matters most in games like this – the story.

Not only is the plot entirely credible, but the universe in which it is set has its own unified system of rules and regulations. In *Ultima*, there is a complete world with its population having their own particular niche whether it be the innkeeper selling you a foaming tankard of ale or a demon lord doing his best to annihilate you in the depths of a foul dungeon.



```
alberich  →     53G
Shamino         57G
Iolo            63G
Gwenno          55G
Toshi           22G
Julia           54G

F:62            G:74
      4-9-139

:INQUISTOR

"This is the
Court of
Inquisition of
His Majesty
Blackthorn!"

"Art thou here
to confess?"

You respond-
:
```

With each successive game, the attention to detail has increased considerably, all the time giving the player a lot more to think about. This is achieved by doubling the number of graphic 'tiles' available. These are the units that go to make up the structure of the land as seen on screen. It is not simply a case of making the existing graphics more detailed, but of introducing a host of new objects. And once those objects have been introduced, then the game is designed so that they have a reason for being there.



Far from your mind while strolling through the nearby woods is the rumor of a mysterious gypsy wagon that some claimed to have seen. But then you see the colorful wagon sitting in the valley below. Strangely, you find yourself drawn to the wagon. As you approach, a woman's voice from inside says, 'Enter, O Seeker.'

Upon entering, you find an old gypsy sitting in a curtained room. In front of her is a table upon which sit two incense bowls. The ancient gypsy smiles as she says, 'We have been waiting such a long time, but at last you have come. Sit here and I shall read the path of your future.' She throws some powder into the bowls that causes a thick blue smoke to boil upwards from them. Staring into the smoke she says, 'Let us begin the casting.'

Thou art an elderly, wealthy eccentric. Thy end is near. Dost thou A) donate all thy wealth to feed hundreds of starving children, and receive public adulation; or B) Humbly live out thy life, willing thy fortune to thy heirs?

Rooms now have furniture. You can sit down at tables, sleep in beds and play the harpsichords. These are all nice touches but they go further than that. If you can find the right person to teach you how to play, and discover what to play and where to play it, then one of the harpsichords will take you to a part of the game otherwise inaccessible. Single trees now adorn the landscape in towns. Have a look in them. You might find something interesting hidden in a hollowed-out tree. Everything is there for a purpose.

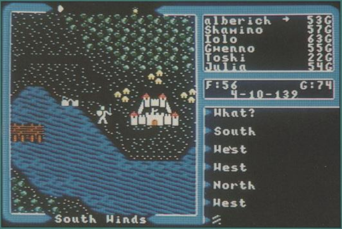
As to the story and your quest, that is left deliberately vague. All that you really know is that Lord British has discovered a whole new Underworld in Britannia. Unable to resist the urge to explore, he manages to get himself captured. As you managed to save Britannia once before *(Ultima IV)* you are summoned back to try and rescue him. But there are other forces afoot too. The virtues that you practised as you strove to become an Avatar are now being misused and a new tyranny is encompassing the land.

As with *Ultima IV*, the determination of your character is achieved by answering assorted moral dilemmas rather than the random rolling of dice. There is no right or wrong answer, you can only reply as you see fit. The game starts with you are two friends in a small hut deep within the forest. Other characters can be recruited as you meet them, up to ten of them although only six can be in your party at any given time. It is up to you to decide who is best suited to each particular task.

Running through the game, you will find that there are a series of mini quests that need to be completed. Rewards can vary – treasure, information or goodwill but you would be well advised to attempt them as each, in their own way will go some way to furthering your information. The other way of finding out what is going on is by talking to people. There are over two hundred characters in the game who have important snippets of information. You won't necessarily learn all this though on your first encounter. Frequently you will told to go and ask someone else about a specific subject so that there is a lot of backtracking to be done.

Another added feature of the game is the passage of time. Rather than stand around waiting for you, the inhabitants of Britannia all have their normal lives to lead. After a busy day working the fields, they will return to their beds and fall into a heavy sleep so that if you arrive at night, you can either go and do something else or wait for them to wake up. Of course there are some people who only wander about at night. You might wonder what they are doing but they will probably ask the same of you! Taking notes is definitely an essential part of the game.

As you cross and recross the country, so you will have to protect your band of merry men from the many groups of monsters roaming the countryside. Combat has been improved so that you can now aim your weapon where you please providing that the target is within range. For example, bows can fire up to 70 feet away, slings only 40 feet. Not all monsters carry treasure but those that do always lug it around in chests that are always locked and frequently trapped. A high dexterity is of use here or else a good supply of unlocking spells. Apart from gold, you may find weapons, food, keys and torches.



```
alberich →  53G
Shamino     57G
Iolo        63G
Gwenno      55G
Toshi       22G
Julia       54G
F:56        G:74
   4-10-139
What?
South
West
West
North
West
```

The magic system in *Ultima V* is still one of the best around. Most characters have some form of spell casting ability in the form of spell points. Before a spell can be cast, its reagents first have to be mixed and note, you cannot do this in the middle of combat. There are eight different ingredients; blood moss, spider silk, black pearl, garlic, ginseng, sulphurous ash, mandrake and nightshade. All but the last two reagents are readily available, the others have to be found. As you can guess, all the important and powerful spells require a pinch of one of these two reagents.

Another nice aspect of the magic system is the variety of spells available. Whereas many games use the same spell in several guises for example small, medium and large zap spells, this, with only a couple of small exceptions is not the case here. Instead, you can change the wind direction, summon elementals, negate magic, create poisonous winds and clone creatures together with forty other spells.

Again, it comes as no great surprise to find that the game is beautifully packaged. Apart from the eight sides of disk, there is a book of lore, command summary booklet, quick reference card, pages from the diary of Lord British, a map of the world come tea cloth and a small coin.

Control of the game is not easy though and it will take even the keenest role playing fan some time to get to grips with things. Commands are entered by a series of keystrokes – A for attack, C for cast and so on. Frequently, a direction has to be specified as well. All this takes time to get used to. You can also designate a given member of your party to carry out all the given actions such as talk, open chests, cast spells and so on. This is fine providing that you have one person capable of doing all these things. Otherwise, you must specify at each choice, who is going to carry out a particular task. All this complexity means that the game is not really suitable for out and out beginners – go and cut your teeth on something a bit easier and then try this.

If I have one grumble about the game, and I must admit that it spoilt it for me somewhat, it is this. Because there is so much extra detail in the game, the program frequently has to access a disk and this slows things right down. All the furniture etc has to be searched and waiting for a response to load in each time seems to take forever.

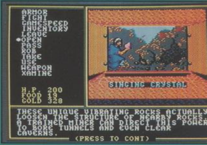
This is one of the disadvantages of an eight bit system used on a sixteen bit game. If you can live with that though, then I am sure that *Ultima V* will you give hours (estimated 100-200 hours playing time) of challenging enjoyment. If you are at least partially interested in role playing games, then Ultima V has got to form a part of your collection.

Title: Ultima V
Supplier: Origin/Microprose, 2 Market Place, Tetbury, Gloucestershire, GL8 8DA
Tel: 0666 54326
Price: £19.95

The Legend of Blacksilver

It was a strange dream. Imagine the princess Aylea coming to you, a mere serf, and claiming that it was you and no other tha had to save the world from the Evil Baron Taragas. It was certainly no dream though when you woke up and found a falcon's feather in your hand.

The problem with Baron Taragas is that he has found Blacksilver in his mines, a strange black mineral that is the source of all magical power within the lands of Thalen and Maelbane. Access to all this power has corrupted



```
ARMOR
FIGHT
GAMESPEED
INVENTORY
LEAVE
•OPEN
PASS
ROB
TAKE
USE
WEAPON
XAMINE

H.P. 200
FOOD 19
GOLD 328
```

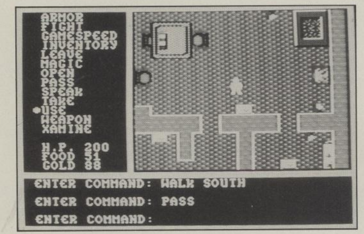
THESE UNIQUE VIBRATING ROCKS ACTUALLY LOOSEN THE STRUCTURE OF NEARBY ROCKS. A TRAINED MINER CAN DIRECT THIS POWER TO BORE TUNNELS AND EVEN CLEAR CAVERNS.

(PRESS TO CONT)

▶▶▶▶▶

the Baron. Princess Aylea's father, King Durek led an army to try and rid the land of this evil once and for all but he was kidnapped from the middle of his camp and has not been seen since. The good wizard Seravol has discovered that this is no longer a quest for armies, but for one man working alone. That still does not explain why the princess chose you but you feel hounour bound to try and help her.

Considering the nature of your quest, you are not given any great favours to help you on your way. Clothed only in shoddy leather armour and with but a handful or gold pieces and a few days food, you set off into the surrounding countryside, ready for whatever adventures lie ahead.



```
ENTER COMMAND: WALK SOUTH
ENTER COMMAND: PASS
ENTER COMMAND:
```

A town is the first place to find, if only to buy yourself a weapon of some sort. The countryside is full of creatures intent on making you their breakfast and many times you have to run away from battle, surely something no would be savour of the world should be doing. Each type of terrain has its own particular set of monsters including quasit hounds, brain spates, bantogs, screaming nugs and night howls. The creatures that inhabit the dungeons are supposedly so bad that no one has survived long enough to name them. There are a few fellow travellers who will help you on your way, peasants, merchants and the like but they never seem to be around when you want them.

Once in a town, you can buy yourself armour, weapons and food. Some places offer banking facilities. Others allow you to gamble your gold away- or make a fortune. There are jobs to be done if you need to earn some extra cash. Prisoners are often a good source of rumours but you will have to bribe the guards first. There are boats to be bought as you progress further and magical spells to help yo in your quest.

Outside the towns, look out for the temples. Here you can participate in mini arcade games to improve your skill levels, cure your wounds and gain access to the archives. These are a series of displays scattered round a maze and are crucial to solving the game. The only problem is that you must have found the appropriate jewel in order to be able to access a particular display. Other places to explore include a castle, citadel and labyrinth, as well as several dungeons.

Entering different commands is simply a matter of choosing from a menu on the left hand side of the screen. Any other instructions such as those required for gambling and training are given on screen as and when you need them. The graphic display is the familiar top down approach with a 3-D display for the dungeons although it must be said that the dungeon monsters are superbly portrayed.

The Legend of Blacksilver is the follow up to *Legacy of the Ancients,* previously released by Electronic Arts although the gameplay and plot have been considerably expanded. As far as complexity goes, the game ranks somewhere in the middle of the multitude of role playing games currently available. *Blacksilver* is highly enjoyable if not over taxing on the old grey cells.

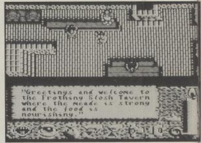
Title: The Legend of Blacksilver
Supplier: Epyx/US Gold, Unit 2 + 4
Holford Way,
Holdord,
Birmingham
B6 7AX
Tel: 021-356 3388

Times of Lore

Anyone glancing casually at this game might be forgiven for thinking that here was yet another *Gauntlet* clone. But then you notice that the game has been written by Origin Systems, they of the Ultima series and surely they would never release anything as trivial as another version of That Game. For I suspect that comparisons with G******t, although inevitable, are the last thing that the author's want to here.

In fact, the game bears considerably more resemblance to *Faery Tale,* the classic graphic adventure from Micro Illusion, so far only available for the Amiga.

Twenty years ago, King Valwyn was busy driving back barbarian hordes. Legends spoke of his return after this passage of time but there is no sign of him and the land is beginning to degenerate into chaos. All sorts of mercenaries are turning up to accept various commission as they seek their own particular goals, treasure or glory. Thus it is that you turn up in the land of Albareth, your sword for hire.


"Greetings and welcome to the Frothing Slosh Tavern where the meade is strong and the food is nourishing."

The game has been designed as one that you can pick up and play almost immediately, there are no daunting 96-page manuals to be studied first. Controlled entirely by joystick, there are a series of icons at the bottom of the screen that allows you a basic form of manipulation of objects and people.

There are many people wandering round this huge game (13,000 different screen locations, 200-300 hours playing time) and you can gain many vital clues by talking to them. You can start chitchat or ask them about any of the rumours that you may have discovered. Some characters may offer additional facilities such as lodging for the night or food for sale.

The other icons are an inventory, a limited examine facility, pick up and drop, use, offer and the game options, pause, score and load. Your position is saved automatically whenever you bed down for the night.

Magic in the game is restricted to scrolls and positions that you find as treasure. These are colour coded and you have to experiment with in order to determine their effects. Only one of each type of artifact can be carried at any given time.

Combat is simply a case of standing next to a foe and stabbing him. This is not as easy as it sounds, for not only is your enemy moving around, but you also have to watch your back in the case of a group of monsters. Most creatures require two good hits before they succumb to your sword. Your strength on the other hand is determined by a melting candle on the right of the screen. Potions will help to restore this as will finding an inn for the night. You must also purchase sufficient food to keep the wolf from the door on your travels. Defeating a monster may lead to him dropping treasure of some sort. Keep a close eye on the screen to see if a small coloured parcel appears.

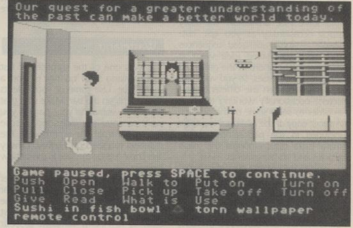
The game consists of a series of mini-quests, each one taking you a step nearer finding the king. Although you can wander wherever you want, you are only going to progress if you stick to the task in hand.

Times of Lore is easy to get into and fun to play if not particularly brain taxing. One word of advice though. If you get some way into the game, back up your disk. Only one game can be saved on the game disk and this position is wiped if you decide to start afresh. **GRH**

Title: Times of Lore
Supplier: Origin/Microprose, 2 Market Place, Tetbury, Gloucestershire, GL8 8DA.
Tel: 0666 54326

Zak McKracken

An early contender for the wackiest game of the year has got to be *Zak McKracken and the Alien Mindbenders* from LucasFilm Games. The only people not likely to find it amusing are those who believe that everything that they read in the *Sunday Sport* is true.



```
Our quest for a greater understanding of
the past can make a better world today.

Game paused, press SPACE to continue.
Push    Open     Walk to    Put on     Turn on
Pull    Close    Pick up    Take off   Turn off
Give    Read     What is    Use
Sushi in fish bowl  ▲      torn wallpaper
remote control
```

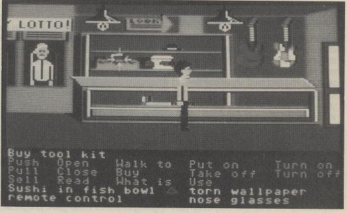
The *Sunday Sport* is not too far from the storyline either. The American equivalent of the SS is the *National Enquirer*. In Zak, you, the eponymous hero, are a reporter on the *National Inquisitor* filling in time by making up stories before going off to win a Pulitzer Prize.

One night, you have a dream. Yes there really is an extra-terrestrial plot. This is one story you don't have to invent.

There is a plot. A plot so dastardly and totally unbelievable that even your editor wouldn't print this one. Aliens have invented a stupidity machine. They have also taken over the telephone company and are busy sending special frequency waves down the phone lines that are reducing everybody's IQ to somewhere approaching zero.

Naturally, as everybody increases in stupidity, no-one believes you. No-one that is apart from Annie, head of the society of Ancient Wisdom and Leslie and Melissa, two girls who are currently on Mars, having been shown how to convert their old beat up van into a modified space rocket.

It is difficult to explain any more as to a large extent the plot is spoiled by revealing it. Also, I haven't got the foggiest idea what I am supposed to be doing next. Suffice to say that funny disguises and two headed squirrels play there part as you travel to, amongst



```
Buy tool kit
Push    Open     Walk to    Put on     Turn on
Pull    Close    Buy        Take off   Turn off
Sell    Read     What is    Use
Sushi in fish bowl  ▲      torn wallpaper
remote control             nose glasses
```

other places, Katmandu, Atlantis and Mars!

The game is controlled by selecting commands from a menu and tying them up with the appropriate verb. Nothing too difficult here although some of the language is occasionally forced by the paucity of available verbs.

As you select your action, so Zak acts it out on screen. This part of the game works well with good graphics and some nice animation. The story stops occasionally as you get 'cut-scenes' – short animated sequences that give you some idea what is happening elsewhere in the game. As the game progresses, so you will have to swap between the different characters, occasionally working together in order to solve some of the puzzles. A spoof copy of the *National Inquisitor* included in the packaging provides many extra clues amongst the inane articles and advertisements.

Purists will object that this is not a serious enough game. Because the plot is so silly, you are forced to try all sorts of ridiculous combinations of objects in the hope of finding something that works. If that is your attitude, then give this game a miss because I can guarantee that you won't like it. Personally, I thought the game was like a breath of fresh air. It is highly original and it made me laugh and there aren't many games that have made me do that – well, not intentionally anyway. But then I like the *Sunday Sport.*

Title: Zak McKracken and the Alien Mindbenders
Supplier: Lucasfilm Games

Hints and Tips

Legend of Blacksilver

Start off by taking the feather to the castle and using it to gain access to the prince. Open all the chests and take the blue gem. Visit the temple and pay to enter the archives. View the singing crystal exhibit and use the crystal that you are given to clear away the rubble in the castle.

Bard's Tale III

Say 'Tarjan' to the priest to gain access to the catacombs. There are only two levels of dungeon to explore here and beginners should build up their party's experience on this, the only easy part of the game. The word that you are looking for is 'chaos' and saying this to the priest takes you to an entirely new dungeon system – Unterbrae. Useful words here are blue, shadow and sword.

As a general tip, use magic items when you find them. There are always more to be found and if you save them, you may well find them useless against tougher opponents. The crystals are used to regenerate lost magic points.

Wasteland

This was one of the surprises of last year, arriving with no bar to find a useful clue. Use some of your talents if you climb on the bar. Make sure that there is a woman in your party if you want to enter the ladies loo. Rescue the mayor from the courthouse. The numbers tattooed on Hewey, Dewey and Louie are for the safe in the outlaw hideout.

In Needles, you need to discover the real bloodstaff which is at the far end of the checker board. Count the number of steps that you make whilst crossing it. Finally, someone in your party will definitely have to learn how to repair toasters!

Dungeons and Dragons

This is still an excellent game despite a ridiculous ending. Considering that the game is called *Pool of Radiance*, I found no reference to it whatsoever throughout my peregrinations. Just before the final battle, there was a picture of the big baddy with a pool behind him. On defeating him though, you just get whisked back to the town council who thank you and then promptly disband leaving you to wander round further if you so choose.

These are mostly general tips as the game plays so differently for everybody and you do not need to complete every task in order to finish the story.

Make sure that you only use weapons that do the most damage for example a morning star is better than a mace. Be careful though with magic weapons. A +1 mace is now better than an ordinary morning star. Although the damage is now the same, you have a better chance to hit and can also use the weapon against all magical creatures.

It is worth paying to identify all magical items even if you are then going to sell it. As the game progresses, there is little else to spend your treasure on and indeed, you will find that it becomes something of a hindrance later on as it slows down your characters too much. Towards the end, I was having to leave behind tens of thousands of gold pieces won by defeating the giants.

The only spells really worth learning are sleep, cure light wounds, detect magic, hold person, fireball and lightning bolt. Other spells occasionally used were knock, read magic and restoration. This last spell is only found on scrolls and is used when one of your characters loses an experience level or two at the hands of ghouls, wights, wraiths and vampires.

The dragon in the dragonspine mountains is friendly and will give assistance. Kuto's Well is worth investigating and should prove a useful place to rest and recover later on in the game. There is a boat hidden to the north of Lake Kuto that will take you to the island but be careful, the pyramid on the island is not an easy place to escape from – a maze with teleports plus several nasty creatures to battle with afterwards.

Whereas fireballs affect a large area, lightning bolts only work in straight lines. A useful tactic is to line up your party and wait until there are monsters in front of everyone. Then send your spellcaster to join the end of the line of monsters and fire away.

Mirrors may prove useful when searching the library and also in Valjevo Castle, especially if you do not want to be petrified. Make sure that you search all the bookshelves carefully.

Whenever you are given a lot of experience points for defeating a monster or finding a treasure, cast a detect magic spell. Some places to look for treasure are the stable in the slums and through an illusionary wall at the back of the keep on Thorn Island.

Finally this month, if you are really stuck in one of Electronic Arts' role playing games, then help is at hand in the form of one of their clue books. These are quite expensive at five pounds each but they are beautifully produced and come complete with all necessary maps and a commentary designed to take you through every step of the game. Titles currently available are *Bard's Tale I, II* and *III, Wasteland, Deathlord* and the *Mars Saga.* The books are available from the Customer Services Department, Electronic Arts, 11-49 Station Road, Langley, Berkshire SL3 8YN tel 0753 46465. ■

Sector Secrets

There's more to disk storage than meets the eye

By Kerry Fowler

Seasoned disk users may be familiar with the concept of track and sector storage but what does a sector contain? If you think that it is just a 256 byte block of data, you are misleading yourself. The sector may be small but its structure is surprisingly complex.

Even the way that information is stored on a disk is different to the normal binary system which the computer employs. Group Coded Recording (GCR) is the system employed by the 1541 and this has been Commodore's standard for the past decade. GCR uses a numbering system which is superficially eight-bit binary but differs in many important respects.

To convert a number from decimal to GCR, it must first be expressed as a binary value:

211 = 11010011

The binary value is then split into two nybbles:

1101 0011

Using a look-up table (see the GCR Equivalence Table), these nybbles are converted into five bit values:

11101 10011

These two values are then combined to give a ten bit value:

1110110011

Then the first eight bits are split off to form a byte and the remaining two bits form the highest two bits of the next byte formed by the following number:

11101100 11......

For example, a series of bytes could be the code for storing a value in zero page:

Mnemonic	Dec	Hex	Binary	GCR
LDA	169	A9	1010 1001	11010 11001
$41	65	41	0100 0001	0110 01011
STA	133	85	1000 0101	01001 01111
$FA	250	FA	1111 1010	10101 11010

When combined the GCR series would look like this:

11010110
01011100
10110100
10111110
10111010

Note that four bytes of normal binary exactly equals five bytes in GCR code. In this way 256 bytes of code would be represented by 320 bytes in GCR, which is all very well but why is it necessary to go to these lengths?

No combination of GCR values will result in a series of eight ones or eight zeros and this is the all-important point. As in most articles of this sort, what the point is is not immediately clear and nor shall it be until the sector is studied in greater detail.

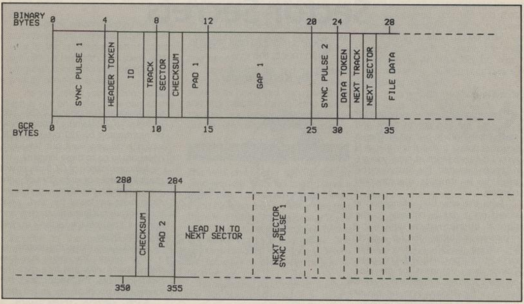
The diagram shows the anatomy of a single sector. The sector has two distinct parts separated by a gap. The head identifies the sector and the longer tail is principally the stored data.

Each section starts with a synchronisation pulse which in both cases is a series of five bytes with the decimal value 255. This poses the GCR system with an anomaly because eight consecutive ones are impossible in GCR. Consequently, the operating system knows that this must be a sync pulse.

This is an essential requirement for the disk organisation because the read head of the drive can land on any part of the track. It must then be able to quickly orientate itself so that it can find the sector it requires and this is where the sync pulse comes in. As soon as a series of 40 ones is encountered a sync pulse has been located. The next question is how does the system know whether this is the head sync or the tail sync?

After the sync pulse there is a token byte. When this 10 bit block of GCR is converted to normal binary, a value of eight signifies a heard block but a data block token has a value of seven. In this way, the operating system knows exactly what it is reading.

What it doesn't know yet is whether the sector is valid. When a disk is changed, the block allocation map (BAM) and directory are not updated automatically. If the disk is accessed without the directory being read. or the drive being initialised in some other way, the new disk could be completely fouled up. You may have wondered why all the manuals insist



that a different ID should be specified for every freshly formatted disk, well here's the answer. The next two sector header bytes (20 GCR bits) contain the ID. If the sector ID does not tally with the ID stored in the BAM any operation will be aborted but if a match is made then the operation will continue according to the information held in the BAM. Offenders who duplicate IDs and then forget to initialise the drive can only blame themselves for the mess that follows.

Up to now the drive knows that it has the header of sector on the correct disk but it has only made a guess at the approximate track position and has no idea at all of which sector it is reading. To confirm the correct track, the next 10 bits contain the track number and a further ten hold the sector number. As a secondary check the operating system has been sequentially applying EOR logic on the IDs and track and sector values. The result is compared to the next byte which acts as a checksum. Now the system is positive that all is okay and it can proceed to the second part of the sector.

The header data is padded out by 20 null bits of GCR code (two bytes when converted) because an exact number of converted bytes is required. So far, in GCR, there have been 40 bits of sync and 10 bits for the other 6 pieces of information which means 100 bits in all. This is equivalent to 12.5 bytes of GCR code and 10 bytes of binary. By adding 20 null bits the ratio becomes exactly 15 bytes GCR to 12 bytes binary which is ideal because the system can only cope with complete bytes in both systems.

The drive spins at about 300 rpm which is far too fast for all of the calculations to be done in realtime. To get around this problem the values are read and stored by the system. After the header there is a gap of 10 bytes of GCR code which allows sufficient time for the data to be evaluated and for the decision to be made whether to read or ignore the next part of the sector.

Assuming that the sector is the desired one, the next sync pulse is registered and the token confirms that it is indeed the second sync. Then follows the data block which is stored in a buffer. The first two bytes of the block are the indicators for the next track and sector in the chain and a checksum at the end confirms the validity of the reading. Once again the total of GCR bytes is not a whole number (40 bits sync + 10 bits token + 2560 bits data + 10 bits checksum = 327.5 bytes GCR) so a further 20 bit pad is added to give 330 bytes GCR or 264 bytes binary.

This means that a sector is 355 GCR coded bytes long (284 bytes binary) and all this is dealt with in a fraction of a second. Who said that Commodore drives were slow?

Between each sector there is a gap denoted by an indeterminate number of zero bytes. The size of the gap varies according to which track the sectors lie on.

This then is the hidden secret life of the sector. It may seem extremely complicated but it does result in an extremely accurate and reliable storage medium. Problems that occur can nearly always be traced to the more delicate mechanics of the drive, the weakest link in the chain.

GCR Equivalence Table

DEC	HEX	BIN	GCR
0	00	0000	01010
1	01	0001	01011
2	02	0010	10010
3	03	0011	10011
4	04	0100	01110
5	05	0101	01111
6	06	0110	10110
7	07	0111	10111
8	08	1000	01001
9	09	1001	11001
10	0A	1010	11010
11	0B	1011	11011
12	0C	1100	01101
13	0D	1101	11101
14	0E	1110	11110
15	0F	1111	10101

DON'T GET LEFT OUT... GET IN ON THE ACTION

COMMODORE DISK USER is a lot more than just another computer magazine. Every issue carries a diskette containing more than £30 worth of software ranging from serious programming utilities to arcade games. There are plenty of Commodore magazines on the market, but we believe that this is the *first* to cater for disk users of all ages and tastes.

COMMODORE DISK USER is what you have been waiting for – take out a subscription TODAY!

SUBSCRIPTION RATES

£15.00 for 6 issues U.K.
£18.00 for 6 issues EUROPE
£18.20 for 6 issues MIDDLE EAST
£19.30 for 6 issues FAR EAST
£18.40 for 6 issues REST OF WORLD
Airmail Subscription Rates on Request

Send your remittance to:
INFONET LTD., 5 River Park Estate,
Berkhamsted, Herts. HP4 1HL.

Please begin my subscription(s) to COMMODORE DISK USER with the .... issue
I enclose my cheque/money order for £.................... made payable to Argus Specialist Publications Ltd.
or debit £.................... from my Access/Barclaycard No. ....
valid from .... to ....
NAME (Mr/Mrs/Miss) ....
ADDRESS ....
....
Postcode ....
Signature ....
Date ....

Please use BLOCK CAPITALS and use post codes.

AMIGA
ATARI ST
IBM PC
APPLE MAC
MICRO Music
THE SPECIALIST MUSIC MAGAZINE FOR COMPUTER USERS
KCS-Just what the Doctor ordered!
MUSIC X-The Review
MICRO MUSIC
Find out about the magazine the computer music world has been waiting for
ACORN
COMMODORE 64
AMSTRAD CPC
ZX SPECTRUM